Volume 1 Nº 2 November 1987 £1.20

Archive

The Subscription Magazine for *Archimedes* Users

Using BBC ROMs

WIMP manager explained

Using Wordwise Plus

Epson Screen Dump

- Zarch is here!
- *HELP Information
- DIY Memory Up-grade
- The RS423 Saga
- Programmers' Quiz
- Lots more Hints & Tips
- Mode 7 Screen-save
- *CONFIGURE explained

Archive is edited by Paul Beverley and published by Norwich Computer Services

Lord, you establish peace for us;
all that we have accomplished
you have done for us.

Isaiah chapter 26 verse 12

Archive

Volume 1 • Nº 2 • November 1987

Contents

News, News, News.....

Archimedes price cut!

As we mentioned in the letter which subscribers should have received with the first issue of Archive, **Acorn has dropped the price of the A300 series by £115** – very nice for those who have not yet taken the plunge and it is good for the future marketing of Archimedes and allied products, but it is a bit frustrating for those of us who have already made our decision to get involved with Archimedes.

Now PC compatible!

The other announcement which came at the PCW Show was that they were offering a "PC compatible version" – The 310M for just £60 (+VAT) more than the straight 310. For the extra money you get a software PC emulator running MS-DOS 3.21 enabling you to run "almost all MS-DOS applications" including things like Lotus 1-2-3, GEM, MS Windows, Word, GWBASIC, Sidekick, Supercalc4 etc, but the software must be on 3.5" 720k IBM disc format and must not have software protection that accesses the disc controller directly. If you have a hard disc, you will be able to partition it for both MS-DOS and Acorn ADFS use. The forthcoming PC hardware podule will widen the scope of PC compatibility further as it will include a 5.25" disc-controller, but it probably won't be available until the first quarter '88.

If you missed out by buying your 310 too soon, you can buy the PC emulator for £89 +VAT.

Available now... The second floppy disc for the Archimedes is available now, price £125 as are the podule backplane (£39), the I/O podule (£79) and the 305-310 RAM upgrade (£89). The ROM podule (£59), the 20Mbyte hard disc with podule (£499) and the Midi add-on for the I/O podule (£29) as well as the A440 itself, look like being available late November time. (All these prices are exclusive of VAT.)

On-line Archimedes Conference

The on-line conference system, CIX, run by Magenta Systems, hosts an Archimedes conference where, if you can pursuade your Archimedes to talk to a modem, you can share ideas with other users. Dr Alpiar who has already contributed two excellent articles to Archive is a regular conferee. If you are interested, contact Brian Smith either on CIX (magsys) on 0483-573337 (Quattro modem, 300-2400 baud) or by phone 0483-65895.

1.2 Operating system

Those of us who bought our Archimedes early and have the 0.2 operating system will be disappointed to hear that we will **not** be offered the 0.3 operating system that is currently being shipped with the 305's, 310's and A310M's. We will have to wait until December, like everyone else, for the 1.2 operating system. Mr Michael Page, Acorn's Corporate Communications Manager said that the early machine were made available mainly for software developers and not for the general end-users and that this was made clear from an early stage.

Well, once again, we shall just have to be patient with Acorn and look forward to receiving the 1.2 operating system, hopefully in December, and hope that it really has had the bugs removed.

The buzz – many of you will have discovered the annoying buzz which occurs after you have pressed <escape> persisting for several seconds. The answer from Acorn is that there is a simple "field change" available to cure this – a small capacitor soldered across a couple of tracks on the printed circuit board. Dealers will carry it out for you on warranty, or, if you are competent at soldering, you can do it yourself – drop us a line we'll send you details.

But, are there two buzzes? The situation is confused as to whether the buzz which distorts the sound output on some machines has the same origin and thus the same cure as the other buzz. Let us know if you have any experience of this.

The RS423 bug(s) – the picture here is a little confused, but it seems that there may still be some hardware problems with the serial I/O chip on some machines, but there is also a software problem. (See article on page 14 for details.)

Software and Hardware for the Archimedes

We continue our list of what is available. This list is in addition to what was published last month. All the information about names, addresses and telephone numbers are in the ***Fact-File*** *at the back of the magazine.*

(If suppliers are miffed because their products are not included, I suggest they get in touch with us and give us some details that we can publish.)

Brainsoft are offering an RS423 connecting lead plus software for £14 which allows you to transfer programs, data files, w/p files etc from a BBC to the Archimedes. This is done "by turning the BBC into a disc drive controlled from the Archimedes"!

Video Digitiser

Those of you who went to the PCW Show would have seen the impressive demonstrations that Mike Harrison was doing of his Archimedes digitiser which Watford Electronics are marketting. The basic resolution is 512 x 256, in 64 grey levels, though you can go up to 512 x 512 on a stationary object. The speed depends on the resolution, but at 512 x 256 it does approximately 10 frames/second and 128 x 128, 25 frames/second. The low level software driver will be in ROM and will include frame grabbing, previewing, scaling, rotation, X/Y flip, negative and pseudo-colour images, printer dumps and data compression. These will all be available via OS commands and SWI calls, so programs can be written in BASIC or other languages with access to these facilities. Price is "around the £200 mark" and availability is "hopefully the first week in November".

Logo

Logo devotees will be interested to know of two ways of getting Logo going on the Archimedes. The first is **LOGOsoft's Master Logo**. It will run under the 6502 emulator and is available now. Ring Chris Squire on 01-891-0989 for more details.

For a Logo that will make best use of Archimedes' amazing processing speed, you will have to wait for **Logotron's Archimedes Logo** which uses full native ARM mode.

Lindy Beveridge, Logotron's PR agency, writes: "The arrival of Acorn's Archimedes is excellent news for Logo users, and Logotron, whose Logo is the standard for British school with over 50,000 users, now supplies it to run on the Archimedes, taking full advantage its high speed processing power. Schools already using Logotron Logo will be perfectly at home with the new software but will notice a considerably enhanced performance.

Archimedes Logo has all the features of a mainframe Logo. It is the most powerful debugging environment for Logo with a workspace of potentially up to 16 Mbytes, the largest Logo workspace on a microcomputer. There is a 132 column editor with 1024 x 1024 pixel graphics resolution. Documentation for the new Logo indicates ways in which Logo users can benefit from the power of the Archimedes.

The first generation of 8-bit educational computers was never able to do full justice to the potential of Logo because of limited memory space and relatively slow processors. "These restrictions have limited the take-up of Logo, but things will change rapidly when people see Logo on the Archimedes", says Logotron's technical director, Christopher Roper. The Archimedes Logo was developed jointly by Acorn and Logotron so that it is functionally compatible with Logotron Logo and has all the characteristic extensions.

For further information, please contact Wendy Frazer on 0223 323656."

(Ed: I was a bit suspicious of the 1024 x 1024 pixels and asked Logotron to confirm it. They said that it was done by re-programming the VIDC controller and obviously was using a multi-sync monitor. They said that Archimedes Logo should be available in full release version in "four to six weeks" from now, which would make it the end of November or the beginning of December.)

Computer Concepts' Wordwise Plus is now available for the Archimedes – working under the emulator. £29 (+VAT) for the complete package and £10 (+VAT) for an upgrade for those who already have the ROM version for the BBC. (See the separate article.)

It's not strictly Archimedes, but you may be interested to know that **Intelligent Interfaces** who produced the IEEE interface podule that we mentioned last month, have now launched an **ARM second processor** for the BBC micro. For £999 +VAT you get a 6.6MHz RISC chip with 4MBytes of RAM. The Twin Editor and BBC BASIC are supplied with it. For more details, contact Mr N Ray of Intelligent Interfaces, on 0789 415875.

We have a pre-release version of **Artisan** from Clares Micros – it's a very impressive art package using the WIMP environment. It has tremendous potential and looks as if it should come up to the standard of some of the Apple Macintosh art software, but we are not going to review it until we get the full release version. It is difficult to give a fair review when a number of the features are still missing. **A**

Documentation Available

ArcIndex – A Programmer's Map for the Acorn Archimedes

This is "an index to the facilities provided by the various different interfaces to the Arthur operating system… it is intended as a quick reference for experienced programmers." 17 pages, A5, laser-printed, price £1.10 including postage from Dr P Hazel, 33 Metcalfe Road, Cambridge, CB4 2DB. It is set out in two columns, the second column containing brief descriptions of system operations, arranged alphabetically, in many cases indexed more than once. The first column contains the name of the system interface that provides the operation – this includes SWI's, *commands, OS calls (OS_Byte, OS_Word etc), VDU calls, font calls, events, etc, etc. So, for example, by looking up "bell", you can find that the channel is set by OS_Byte 211, the duration by OS_Byte 214 etc and by checking "OS_Evaluate Expression", you can find that it is SWI 45. There are over 800 entries altogether. This document would obviously be very useful to an experienced programmer, but that lets me out – I'm too busy editing magazines to have time to do much programming!

Archimedes Programmers' Reference Guide should be available about mid-November from Acorn Computers – essential reading for anyone who wants to do any kind of serious programming with the Archimedes. It is intended to be used as a reference work and as such is not exactly bed-side reading. Hopefully the final addition will have what is lacking in the draft edition which I am trying to use – an index! Also, it is apparently now going to come in two parts like the Master Reference Guide.

ARM Assembly Language Programming by Peter Cockerell, published by MTC, Gaddesden Place, Hemel Hempstead, Herts, HP2 6EX. – a fascinating insight into the whole philosophy of reduced instruction set processors and a useful starting point for Archimedes users who wants to program in assembly language – price is £12.95. (£12 to full members of Archive – including UK postage and packing.) – a review follows…

"ARM Assembly Language Programming" by Peter Cockerell

The ARM Assembly Language Programming book is the first book available that is devoted entirely to the new and revolutionary ARM Processor Instruction Set. The book is published by MTC (at the same address as Computer Concepts) and is written in the form of seven chapters with three appendices – the technicality of the chapters gradually gets more complicated as you progress through the book. If however you have previous knowledge of any type of machine code programming, certain parts can be skipped over. The format of the book makes it very easy to use for reference as well as for tutorial purposes.

The first chapter contains the very basics of the ARM structure – the way in which bits, bytes and binary work and the differences between assembly and BASIC as well as details of the "barrel-shifter and ALU". This type of start will be familiar to all who have read any other machine code tutorial book. The book goes on to explain the ARM Processor and its capabilities, its house-keeping and, of course, its instruction set. The chapter regarding the instruction set is split into six groups, condition codes, data manipulation, load and store, multiple load and store, branch and software interrupt instructions. Each group of instructions is summarised, which makes future references much easier.

One feature of Acorn microcomputers is the built-in BASIC assembler which is a welcome facility for all machine level programmers. Chapter 4 discusses the ARM assembler in BASIC V in some detail. The book is written without too many references to BASIC or 6502 which can be confusing for new users although could be of assistance to experienced programmers. The remaining chapters and appendices contain data structures, non-user modes, co-processor instructions, floating point instructions sets and a complete summary of the instruction set.

I bought the book on impulse just before I bought my Archimedes in early August. I have found it invaluable since it is, as far as I know, the only ARM assembly language book available at the moment. It provides a method of exploiting this new exciting technology until the Programmer's Reference Guide finally arrives at the end of November(?).

The book itself costs £12.95 from Computer Concepts and I would recommend it for any computer power addict. (£12 through Archive magazine.)

Matthew Treagus

Hints and Tips

RGB Connector

Hugh Nolan, of Old Roan, Liverpool, writes, "The pinout for the RGB connector on page 18 of Archive 1.1 indicates that it follows that of an IBM PGA rather than the more common EGA standard. If anybody has a monitor supplied with a lead for use with an EGA and wishes to build a new cable they should note that the EGA pinouts are as follows:

1 Ground	6 Secondary Green/Intensity
2 Secondary Red	
3 Red	7 Secondary Blue/Mono
4 Green	8 Horizontal Retrace
5 Blue	9 Vertical Retrace

I am currently using such a monitor with my BBC model B and the connections are fairly straightforward except that the BBC's connector does not have separate vertical and horizontal syncs, so I have wired the composite sync to pin 9 (Vertical Retrace) of the monitor."

Using a multi-sync monitor

A much easier way to change to a multi-sync monitor is to use:
*CONFIGURE MONITORTYPE 1 and
*CONFIGURE MONITORTYPE 0 to switch back to the normal monitor.

Epson screen-dumps

No doubt you have discovered the modules called HardCopyFX, MX and RX, and you too have discovered that they don't work! Gerrald Fitton has the answer for us. His bug-fix has been passed on to Acorn who are also circulating it. The recipe is as follows:

*LOAD Modules.HardCopyFX 10000 (that's four noughts - beware of using three!)

```
!&104E0=&E59C1020
!&104E4=&E20100FF
```

(now replace the disc for the one onto which you want to put the hard-copy module)

```
*SAVE HardCopyFX 10000 + 700
*SETTYPE HardCopyFX &FFA
*STAMP HardCopyFX
```

To use the module, *RMLOAD HardCopyFX (or just *HardCopyFX will do) and then call it with *HardCopyFX plus various parameters – *HELP HardCopyFX will show you what all the parameters are. Has anyone been experimenting with it? What do all the parameters mean and what are their ranges? (Landscape, XScale, YScale, Margin and Threshold)

Avoiding constant use of *MOUNT

If you, like me, have a number of different discs and you can't remember what is on which, you will probably be fed up of having to type *MOUNT each time you change the disc before you can get a catalogue. Try *CONFIGURE NODIR and <ctrl-break> and you will find that as long as you haven't gone down to lower directories, you can just change the disc and do a *CAT (or *. works as on the BBC micro) and it will re-catalogue the disc in the drive. For more details, see James Lynn's article about *CONFIGURE.

File copying problems

There are one or two problems with the *COPY command under the 0.2 and 0.3 operating systems but, not being used to using ADFS myself, I am never sure when it is a bug and when it is me being stupid. One thing you will find though is that when doing a *COPY from disc to disc using a single drive, it seems to hang up when it should be saying "Enter source disc and press the space bar" but don't panic, it's probably still OK and it's just that the prompt has not been printed. Replace the disc you were expecting to put in next and press <space> and I think you will find that it carries on OK. If you haven't discovered how to do disc to disc copying on one drive, see the "Have you discovered" section.

File transfer problems

If you are still having problems getting files across from other computers to the Archimedes, we have received one or two programs from subscribers. There is not enough space to print them in this issue, but if you drop us a line, we could send you a listing.

Attaching a 5.25" disc drive

Acorn have told us that they would not recommend us to do this as the disc drive hardware in the computer is only designed to drive 3.5" drives with low control line loading and that if you want to attach a 5.25" drive, you should have suitable buffer circuitry added. The feedback we are getting for subscribers is that most of you are having success with this, though one person said that one drive he attached would not work whilst another one did and another person actually blew up the driver chip in the Archimedes. So, it's up to you, but in the light of these comments from Acorn, it would clearly be a good idea to remove the pull-up resistors on the auxilliary drive.

(For an alternative, check the "Hardware Available" section to see what Brainsoft have done to get round the problem.)

Three floppies on the desktop

In order to get three floppies on the desktop, i.e. assuming you have added a double 5.25" disc drive, Gerrald Fitton again comes to our rescue. "In the BASIC library program called $.desktop.accessory.filehandlr you will find that lines 130 and 140 accept the first and second floppies, so you will need an extra line:

```
145 IF floppies%>2 THEN PROCsys
 _addtoiconbar_left("floppy2",…
```

etc as per lines 130 and 140 and you will have to make a further copy of lines 290 to 420 (which refer to floppy 0) emulating the changes made to generate lines 430 to 560 which refer to floppy 1 by inserting "2" in the appropriate places.

Fortran 77

Tom Fortescue warns potential purchasers of Fortran77 that (a) it needs at least 1Mbyte and is therefore unusable on the A305 (I haven't been able to check this with Acornsoft.) and (b) it does not come with an editor, so unless you've got View or Wordwise or Inter-Word or somesuch, you won't get very far with it!

Stereo speakers

Enoch Mayer says that he has bought, for around £25, a couple of "Realisitic" (Tandy) amplified speakers designed for use with personal stereo radio/tape players. (Model: Minimus-0.6 amplified speaker system Cat № 40-1259E) Although they run off two sets of four size-C batteries, they can easily be adapted to run off small battery eliminators. They tone in nicely with the Archimedes, but the best thing is that when no sound signal is received they automatically shut off, thus eliminating the "fuzz"! A

Questions and Answers

Q: I'm desperate for a word-processor. Is there any way I can use the BASIC editor as a word-processor.

A: Well, it won't set the right number of characters to the line – you will have to do that manually using the split lines and join lines functions – but it is perfectly possible to edit and print text using the editor. If you type in your text and then escape from the editor and type LISTO8 before you list the "program" to the printer you get a reasonable effect. To list it to the printer, you use, LIST<ctrl-B><return><ctrl-C> but beware that there will then be a ">" prompt lurking in the printer ready to be printed out unless you clear the printer buffer. One thing you do have to be careful of is using BASIC keywords and their abbreviations, e.g. if you try to write about the T.U.C. Conference, you will find it referred to as the "TANUNTILCOLOUR Conference"!

Q: Does BASIC V need line numbers or not? One article in Archive 1.1 appeared to say it did, and the other that it did not.

A: You can program in BASIC V without any reference to line numbers – even ON ERROR calls can be dealt with by using ON ERROR PROCerror: END and then having a definition somewhere of PROCerror. So in theory you don't need line numbers. However, in practice, you do need them in order to get BASIC V to accept the lines of code that you have written. If you want to avoid line numbers altogether while editing, you can use Wordwise Plus (See article on Using Wordwise Plus – page 20.)

Q: How do you alter the clock on the Archimedes?

A: To change the time, go into the desktop, select the clock and press the menu button on the mouse (middle button) – the rest is obvious. However, you don't seem to be able to adjust the day and date from the desktop.

To set the date, you use:

```
*SET SYS$DATE WED,21 OCT
*SET SYS$YEAR 1987
```

To check it, type `PRINT TIME$`.

Q: The clock on my computer seems to run slow when I do things within the desktop. Is that right?

A: Firstly, you will find that when using the desktop, if you put up the clock on the screen it will appear to run slow when you use some of the other widows, especially file handling. However, if you check it against your watch, you will find that when there is no other activity, the clock catches up again. In fact I set up the clock on my Archimedes to the nearest second against my own watch and checked it again 48 hours later – it was spot on!

Q: Is there any way I can get a modem working on the Archimedes? – I'm suffering from "Prestel withdrawal symptoms"!

A: The only software I've heard of, so far, that is supposed to work on the Archimedes is Watford Electronics' Modem 84 Prestel Information ROM which apparently, so one user said, works under the 6502 emulator. (See the article on using BBC ROM software.)

Q: Why doesn't *POINTER 2 do anything?

A: It will not work until you define a shape for your pointer. Mr J Shipp has sent us a program for a very realistic hand-shaped pointer. See the program on page 41.

Q: Why is the clock speed of the Archimedes quoted as 4/8 MHz?

A: The clock speed depends on what exactly the processor is accessing. The RAM is full speed and works at 8 MHz whereas the ROM is slower working at only 4 MHz. This explains why the RAM version of BASIC V works faster than the ROM version – not twice as fast because even when the BASIC interpreter is running in ROM, it is accessing the area of RAM which holds the program and an area of RAM for workspace as well as the screen RAM.

Q: How many podules can you have on the Archimedes?

A: Two on a 300 series and four on a 400 series. Some podules are double width but, as far as I can gather, this does not change the number that you can get into each machine. Remember that the 300 series computers needs a back–plane adding before you can add any podules (£39 + VAT) and that the 440 will already have one slot occupied by the hard-disc controller.

Q: Solidisk say they are doing an upgrade for a 305 or 310 to 4 megabytes. Is that really possible?

A: Yes, but the presently available operating system will only cope with one megabyte. Therefore the upgrade will not be usable until there is a suitable operating system available.

Q: I have discovered two undocumented keywords: TWIN and TWINO. What do they do?

A: They are to allow you to edit BASIC programs with the TWIN editor. TWIN converts the program into text whereas TWINO converts it and adds the indentation. Unless you have the TWIN editor, these commands will not do a lot for you!

Q: When trying to use my computer with the desktop it stops and the following characters appear at the top of the screen: "Naff RTC MONTH" but it works OK on all my other software. What's wrong?

A: The characters are presumably an error message, though I can't find it in the Reference Guide, which could be translated "Excuse me, I think the month indication on your real time clock in not working correctly."! I had heard that the real time clock chip sometimes works loose in its socket, so I suggested taking the lid off and firming the chip into its sockets with thumb pressure. This did not do the trick. Beebug's suggestion was to reset the CMOS RAM settings by holding the R key down at power-up (twice, because it toggles the monitor type each time you do it). This had no effect either. Eventually we wondered if the information in the RTC might have been corrupted. Use of *SHOW and *SET SYS$DATE (as mentioned above) soon put it right! We have no idea how it got corrupted, but at least it is now working and the desktop is usable again.

Q: How do I take the lid off the computer?

A: Loosen the two small screws at the sides of the computer, just behind the plastic facia – they are in slotted holes so you don't need to remove the screws altogether. Then the theory is that you loosen the three screws at the back along the top edge of the lid, but I find it easier to take them right out. The lid then lifts up slightly and slides backwards pulling out from under the other two screws.

Have you discovered... ?

We continue our listing of bits and pieces that may well be obvious to some of you, but if you haven't discovered them yet, they could come as a welcome revelation.

HELP with BASIC V – If you load the ram version of BASIC from the Welcome Disc (you can use QUIT and RMLOAD $.M*.R* and then BASIC) you will find that HELP gives all sorts of useful information. For example, HELP . lists out all the keywords and then HELP LISTO or HELP TWIN or whatever will give you information about using that keyword. This does not work with the ROM version of BASIC. If you can't quite remember the spelling of the word you want, type something like: HELP L, and it will give you all the keywords beginning with L.

The LISTO command, familiar from the BBC micro has been extended. As well as allowing indentation on REPEAT and FOR...NEXT loops and putting in a leading space on each line, it now has options to split multi-statement lines (but it does not then indent the structure), list the program without line numbers and print the keywords in lower case. Type HELP LISTO in RAMBASIC to see the different features or look at page 267 of the User Guide.

If the first line of a BASIC program is

```
10 REM > FILENAME
```

then by simply typing SAVE and pressing <return> the program will be saved in the current directory as "FILENAME". To be sure you don't save it in the wrong directory you could put the full pathname something like:

```
10 REM > $.BASPROGS.FILENAME
```

OFF turns the cursor off – rather easier than trying to remember VDU23etc and surprise, surprise... ON turns it back on again!

Copying with a single drive – To copy files from one disc to another using a single drive, add a "P" to the end of the command – P for Prompt. So, for example, to copy all the files in the default directory on one disc that begin with, say, "G", into the "PROGS" directory on the other disc, you would type:

```
*COPY :0.$.G* :0.$.PROGS.G* P
```

and you may also want to make it "PQ" (Q for Quick) so that you don't have to switch discs for virtually every file that is transferred. Adding Q means that it uses the language workspace for the copying which will of course delete any text or program currently in memory. (Note that in the 0.2 operating system, adding Q sometimes causes the system to hang up.)

Another way of copying files from directory to directory or disc to disc, there is provided by a program in the UTILITIES directory on the Welcome disc called DIRCOPY. It is fairly easy to use – if you don't know what reponse to give at any point just press <return> and it will give you some help.

Using wildcards on ADFS – To save typing time and also to save trying to remember the exact spelling, you can select a directory with a name including wildcards, e.g. *LIB L* on the Welcome disc sets the library to the LIBRARY directory.

Similarly, in BASIC, you can load (or chain) programs using wildcards in the name. So if you put in the Welcome disc and type CH. "U*.D*" it will run the DIRCOPY program out of the UTILITIES directory. Or if, like me, you find the quotation marks a pain to find on the keybard, (being in a different place on the BBC which I use along side the Archimedes), you can type in *BASIC U*.D* and it will CHAIN DIRCOPY!

You can also use wildcards for loading files on Wordwise Plus and View.

The use of the menu button on the mouse when using the desktop manager... Pressing the middle, menu, button when on a disc catalogue gives you a menu for renaming, deleting etc. If the words in the menu are in shaded type, as opposed to solid type, it is because you need to select a file or folder on which the commands are to operate.

If you click the menu button on the disc icon itself, it allows you to format the disc.

The menu button used on the clock allows you to adjust the time and on the notepad and diary, it allows you to load, save and print it.

HELP on modules – If you want to find out what a particular module does, one way to get at least a few clues is to clear all modules with <ctrl-break> or *RMCLEAR and then *RMLOAD the module in question – say *RMLOAD $.MOD*.DEB* to load the debugger – then type "*HELP ." (notice that it is *HELP<space><full stop>) and you will get an amazingly long list of HELP information about everything you never (sic) wanted to know, and at the very end is the HELP information about the module you have just loaded. We have prepared an alphabetically sorted list of this help information which can be found on page 24 of this issue. A

DIY Memory Up-grades

All that is involved in the up-grade from A305 to A310 is to purchase and plug in the appropriate chips, so in theory it should be very simple and we intended therefore to publish an article telling you how it is done. However, the up-grade is not as easy to do as it looks since the RAM sockets are underneath the bar on which the disc drive(s) is(are) sitting. Also, life is much more complicated if you have also got one or more podules fitted.

To explain where every screw is and what to do in every eventuality became far too complicated to put on paper without complicated diagrams. If you want to get a dealer to do the up-grade, you will presumably have to pay Acorn's prices (£102.35 incVAT), but if you want to do the up-grade yourself, you can get the chips, for example, from CJE Micros (£88 incVAT) and try fitting them yourself. They will provide "basic instructions" about fitting the chips, i.e. which way round they should go, for example, but not, as yet, the kind of detailed description of how to get the board out (and back in) that we had intended to give in this magazine.

However, it is not that difficult to get the board in and out – the main thing to watch out for is the nasty sharp self-tapping screws that stick up from underneath the base plate. If you are not careful they can rip tracks off the bottom of the p.c.b. as you slide it in and out!

The other problem is that if you do the up-grade yourself, you will presumably be voiding the warranty! A

The *CONFIGURE command

James Lynn

The *CONFIGURE command allows the user to set certain values in the Archimedes non-volatile RAM so that the machine is set up to his or her particular preference. Such things as the default filing system, the amount of memory for the screen etc. can be set using the *CONFIGURE command. As well as the operating system, relocatable modules can have their own CONFIGURE options.

To find out which CONFIGURE options your Archimedes offers, use the command *STATUS which will give the current settings of the options available, or *CONFIGURE <return> which gives the available options and their respective syntax. The Archimedes User Guide lists many of the options, but does not list them all. A fuller list is presented here.

Baud <n> sets the receive and transmit rate for the RS423 interface. The meaning of 'n' is as follows:

Value	Baud Rate
0	9600
1	75
2	150
3	300
4	1200
5	2400
6	4800
7	9600
8	19200

Boot This option causes RESET to act like <shift-break> (and boot the disc) and <shift-break> to act like a normal RESET but <break> still has the same effect as <escape>. The alternative option is NOBOOT.

Caps turns the Caps-lock on after a RESET.

Data <n> specifies the default format for data when using the RS423 interface. The value of 'n' specifies the format as follows:

Value	Word Length	Parity	Stop Bits
0	7	even	2
1	7	odd	2
2	7	even	1
3	7	odd	1
4	8	none	2
5	8	none	1
6	8	even	1
7	8	odd	1

Dir When the machine boots up, this option causes the ADFS to look at the disc in the drive and read in the root directory. The side effect of this is that every time you put in a new disc, the ADFS requires you to tell it to read the directory, with a *MOUNT command, otherwise it will give a 'disc changed' error.

Drive <n> sets the default disc drive (0-3 being floppies, 4-7 being hard discs)

DumpFormat <n> sets the format for *DUMP and *LIST. It determines how control characters are handled. The lowest two bits have the following meaning.

Bit0	Bit1	Meaning
0	0	Standard Acorn format as in *KEY
1	0	Print a full stop for control codes
0	1	<n> is printed where n is in decimal
1	1	<&n> where n is in hexadecimal.

If bit 2 is set, all characters above 128 are treated as printable characters, otherwise they are treated as control codes.

If bit 3 is set, all characters have their top bit stripped off before being processed, otherwise they are unaltered.

File <n> sets the default filing system on reset. The only filing systems currently available on the Archimedes are n=5 for Econet, and n=8 for ADFS.

FileSystem <n>/name If n is given, this option is identical to 'File'. The 'name' option allows you to specify the name of the required filing system, eg 'ADFS' or 'NET'.

Floppies <n> tells the machine how many floppy drives it should think are attached.

FontSize <n> reserves n pages of 4K RAM for the font cache.

FS [nnn,]<sss>/<name> sets the network fileserver. <sss> is the station number. <nnn> is the network number (optional). Instead of the station number, a name can be given.

HardDiscs <n> tells the machine that it should have <n> hard discs attached.

Ignore [<n>] tells the machine which character should not be sent to the printer, where <n> is the ASCII code of the relevant character. If <n> is omitted then all characters are sent to the printer.

Language <n> tells the computer which module number it should select after a hard BREAK. This is only available on machines with a version of the OS greater than 0.30.

Lib 0/1 If 0, then the default fileserver library is used (usually $.Library), if 1 is selected, the library used will be $.ArthurLib. This is only available after OS 0.30.

Loud sets the CTRL-G bell at full volume.

Mode <n> sets the screen mode on reset to mode <n>.

MonitorType <n> tells the machine what type of monitor is attached. Type 0 is a standard 50Hz monitor. Type 1 is a multi-sync monitor, which allows the use of modes 18-20 (which have twice the normal vertical resolution and therefore require a faster scan rate than normal monitors can manage). Type 2 is for a hi-resolution 64kHz monochrome monitor, only with 400 series machines.

NoBoot is the usual arrangement. <shift-break> will boot up a disc, RESET does a soft break, and breakacts like escape.

NoCaps causes Caps lock to be off on reset.

NoDir is the opposite to CONFIGURE DIR. This tells the ADFS not to read in a directory on boot-up. Thus, every time you change discs and try to save or load anything, the ADFS knows this and automatically reads in the directory of the new disc, so you don't have to keep typing *MOUNT every time you change discs.

NoScroll Whenever a character is printed to the bottom right hand character on the screen, the screen normally scrolls. This configure option prevents the screen scrolling until the next character is printed, allowing you to print the last character on the last line without losing the top line from scrolling.

Print <n> defines the printer type. Type 0 is a 'printer sink' meaning that any characters sent to the printer are discarded. 1 is the parallel printer port. 2 is for a serial printer plugged into the RS423 port, 3 is a user printer driver and 4 is the network printer.

PS[nnn,]<sss> or name sets the network printer server number. The arguments are the same as for the FS option.

Quiet sets the volume of the CTRL-G bell to half volume.

RamFsSize <n> sets the number of pages of memory for the RAM filing system module (if present). The page size is 8K on a 300 series machine and 32K on a 400 series machine.

Repeat <n> sets the auto-repeat rate of the keyboard to n hundredths of a second.

RMASize <n> sets the number of RAM pages available in the relocatable module area. Page size is 8K for 300 machines, 32K for 400 machines). For OS before 0.30, <n> is the total number of pages available for all relocatable modules and their workspace, including the workspace needed for system modules. On OS version 0.30 and later, <n> is the number of pages allocated AFTER the system modules have claimed the space they require.

ScreenSize <n> sets the number of pages (8K for 300 series, 32K for 400 series) available as screen memory. The default changes depending on your machine. On a 512K machine, 10 x 8K pages are reserved, giving 80K, enough for all modes. On machines of 1MByte and greater the default size is 160K (20 x 8K pages on 300 series machines, 5 x 32K pages on 400 series machines).

Scroll disables the scroll protect option, meaning the screen scrolls as soon as the last character on a line is printed.

ShCaps sets the Shift Caps Lock option so that unshifted alphabetic keys give upper case characters and shifted alphabetic characters give lower case characters.

SoundDefault <spkr> <vol> <voice> sets the default sound characteristics. The <spkr> argument turns off the internal speaker (0) or turns it on (1). <vol> gives the volume from 0 to 7 (lowest to highest). <voice> sets the voice that will be assigned to channel 1, and is in the range 1 to 16.

SpriteSize <n> reserves <n> pages for the sprite system (page size 8K or 32K).

Step <n> [<drive>] sets the step rate for floppy drive to <n> ms. If <drive> is omitted, the step rate will apply to all drives, otherwise it applies only to the selected drive.

Sync <n> selects vertical sync only (0) or composite (1) sync. The default is 1 and is the only correct setting for the standard Archimedes monitors.

SystemSize <n> reserves <n> pages for the system heap. (page size 8K or 32K) The default amount is 32K.

TV <n>[,<m>] sets the default setting for the *TV command parameters.

The RS423 Saga continues...

First of all, apologies that we have added to the confusion by publishing an **incorrect pin connection diagram** for the Archimedes RS423 connector. Pin 7 should have been RTS, not DTR and pin 8, CTS and not RTS. That was my typing error - sorry about that.

Secondly, in the program for RS423 file transfer, the ***FX156,&56,0 command** was used. This is OK on the Master, but the ordinary BBC B does not like it. In practice, unless you have re-configured the RS423 format on the Archimedes, you can omit the command altogether on both machines.

The other point worth mentioning is that Acorn tell us that some problems can occur due to the fact that a single clock is used to generate both the sending and the receiving baud rates. They say that even if you are only transmitting in one direction, it is best to **set both the baud rates on the Archimedes to the same value.**

Never-the-less, there are some problems not of our making. Firstly, it seems that there are still **hardware problems on the serial I/O chip** on some machines and secondly there is also a software problem.

It appears that the serial chip is a bit "marginal"; in other words you will find that in some machines it works OK and in others, the RTS and CTS control lines just will not work, as is the case on my machine. Acorn are looking into getting alternative chips but in the meantime, they are trying to convince us that most people are able to do all they want to with the RS423. If you are one of those who is still not able to use the RS423 properly and if you are sure that it is not because of incorrect pin connections or because the software is not working properly on the Archimedes, let David Bell, Hardware Products Manager at Acorn, know about it so that he can gauge whether the problem is really serious or not.

On the software side, there was a bug (or two!) in both the 0.2 and the 0.3 operating systems. Those of you with the 0.3 should have a software "patch" on your Welcome disc, but that patch is specific to 0.3 and will unfortunately not work on 0.2!

The experience of a couple of folk is recorded below...

Ian Kirkup, with the help of a few ideas that I sent him, has managed to get **on-line to Prestel**. Part of the problem was getting his BBC ROM software to work under the emulator (see separate article). The other problem was with the connections to his modem. Apparently, he found that the connections, as labelled in the 'Magic Modem' manual, were incorrect and that the RTS and CTS lines on the modem connector are the opposite way round from the ones on the BBC Micro, i.e. CTS is top left and RTS is bottom left looking into the socket.

Dr Ronald Alpiar writes about **using the BBC micro as an I/O podule** via the RS423...

"In the first issue of Archive I promised a follow-up article about an Archimedes 'software I/O podule emulator', which I have regretfully been forced to postpone. The reason for this is interesting, and may be instructive.

Preliminary plans for the emulator appeared to work, and after an interval of doing other jobs on the machine, I returned to RS423 operation to put on the finishing touches. To my surprise the emulator refused to work! Even the 'self-test' (cf last issue) failed on the Archimedes. Consternation and dismay! Check all wires and plugs – no joy – recheck – panic – say a little prayer! Eventually the sad truth dawned...

Having completed my previous article on an A305, the ever-obliging manager of my local computer shop let me part exchange my A305 for an A310: neither he nor I then suspecting that it had a defective RS423 chip. The latter seems to work perfectly on transmit, but refuses to receive even at the slowest speeds.

This reinforces Paul's warnings about the RS423 hard & software, and indicates wide performance variations in the batch of communications chips installed in current Archimedes'.

The promised article will be submitted to Paul as soon as I get this problem sorted out – by which time the more energetic readers will doubtless have devised their own emulators!"

Using the WIMP Environment

Adrian Look

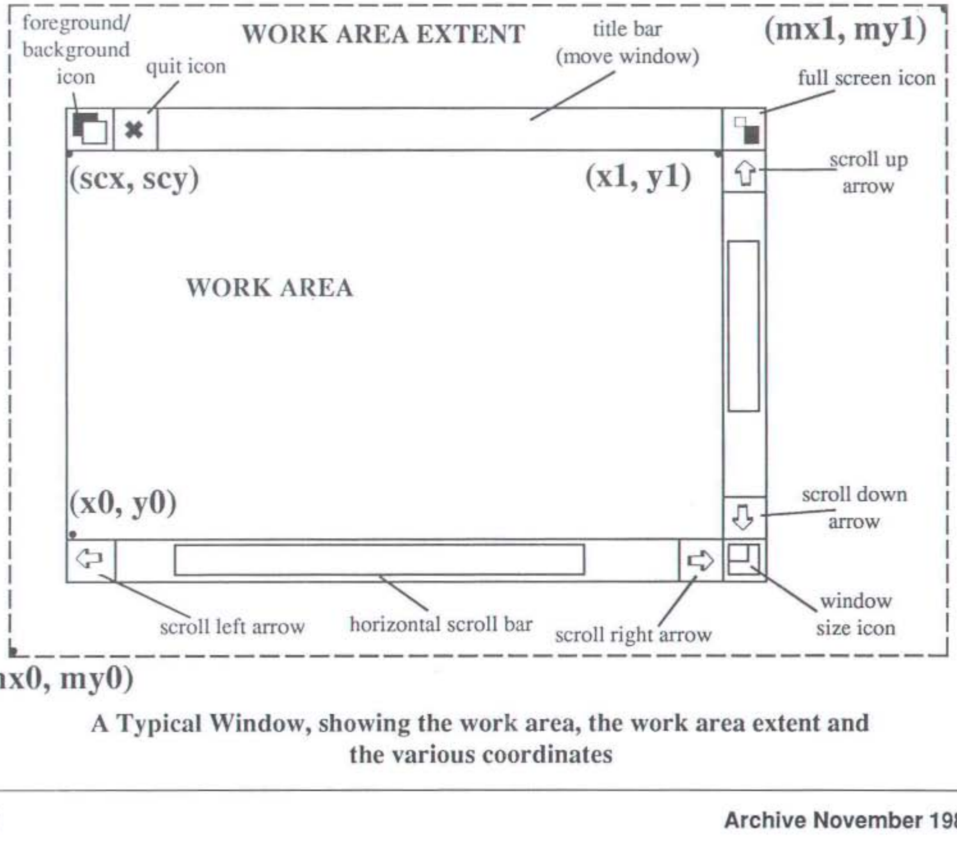
The Arthur Window Manager (AWM) is very complex and contains many high-powered routines. I have decided that instead of studying all the ins-and-outs of the AWM, this series of articles will focus on allowing the user to set-up and operate his own simple WIMP environment in BASIC. I shall do this by producing a set of procedures and functions. These will be accompanied with full explanation of how they work, as well as some of the theory involved.

In this first article we will look at how to set up the WIMP environment. After all we cannot explore the facilities offered to manipulate the windows until we can set them up!

The AWM consists of a series subroutines which may be accessed from BASIC via the SYS command. This command allows the user to access the operating system subroutines – indeed any machine code present in the machine. The SYS command allows data to be passed between BASIC and the machine code routine being accessed. It takes the form:

SYS address [variables] [TO variables]

The address part of the command can, for the operating system, be replaced by a string. eg "Wimp_Initialise" instead of the address &400C0. This makes programs much more intelligible – it's a good thing Acorn put it in. For more than just a brief outline of this command you should read the User Guide.



A Typical Window, showing the work area, the work area extent and the various coordinates

'Wimp_Initialise' at &400C0

This subroutine should be used when the application 'starts up'. It sets up the AWM so that no windows are defined. Thus if any previous programs used the system, the slate would be wiped clean ready to use. This routine also returns the version of the AWM installed in the machine. If the AWM installed is of version 0.07 or earlier then the routine will return a value of 0 for the version.

```
:
version=FNwimp_initialise
:
END
:
DEFFNwimp_initialise
LOCAL version%
SYS"Wimp_Initialise"TO version%
=version%/100
```

'Create_Window' at &400C1

This routine allows the user to set up the configuration of any new windows and informs the AWM of their presence. In response to this, the AWM assigns the window a 'handle', that is a number by which the window is recognised.

This diagram shows the layout of a window, showing the various coordinates used. The work area extent coordinates indicate its size, the work area coordinates are screen coordinates, the scroll bar coordinates indicate the position of the top lefthand corner of the visible area within the work area extent.

AWM windows consist of two main areas, the work area (the visible part of the window) and the work area extent - the full size of the window. The work area extent can be larger than the screen, as all of this area may be accessed by means of the scroll bars. The scroll bars are shown on the sides of the window, and they indicate which part of the work area extent is being displayed (the part we can see on the screen). I shall assume that, having used the Welcome disk, you know how to operate the windows' facilities. If you aren't, look it up in the Welcome Guide (pages 18-24).

The 'Create_Window' routine allows the user to set up a window to his (or her, understood!) own specification by creating a block of information on the said window, then calling the routine.

The data required in the block:

- Work area extent coordinates (mx0,my0, mx1,my1) - the whole window area
- Work area coordinates (x0,y0,x1,y1) - the visible part
- Scroll bar positions (scx,scy)
- Colours:- title, foreground & background
- Work area, foreground & background
- Scroll bar, outer & inner
- Highlight – if the window is inputting text, its background gets set to this colour
- Flags:-Has title bar
  Is moveable
  Has a vertical scroll bar
  Has a horizontal scroll bar
  Has no 'back' or 'quit' boxes
- Title bar type:- Has a border
  Contains text
  Text is horizontally-centred within box
  Text is vertically-centred within box
  Has a filled background
- Title String:- Twelve characters at maximum
- Stack_position:- where the window is placed in priority of the AWM window list (-1 is the top, -2 is lower etc) the top should be the default value. ie -1

There are other options which I have not included at this stage but, as they become relevant, I shall reveal them in later articles.

```
DIM block% &100
    :REM space for data block
handle%=FNcreate_window(block%)
    : REM calling the routine
END
:
DEFFNcreate_window(block%,
                 ...variables...)
LOCAL block%,handle%
PROCset_up_block(block%,)
SYS "Wimp_CreateWindow",
                block% TO handle%
=handle%
:
DEFPROCset_up_block(block%,
                 ...variables...)
LOCAL block%
:
block%!0=work_area_x0
block%!4=work_area_y0
block%!8=work_area_x1
block%!12=work_area_y1
block%!16=scroll_bar_position_scx
block%!20=scroll_bar_position_scy
block%!24=stack_postion
block%!28=FNflags(...variables...)
block%?32=title_foreground
block%?33=title_background
block%?34=work_area_foreground
block%?35=work_area_background
block%?36=scroll_bar_outer
block%?37=scroll_bar_inner
block%?38=highlight_colour
block%!40=work_area_extent_mx0
block%!44=work_area_extent_my0
block%!48=work_area_extent_mx1
block%!52=work_area_extent_my1
block%!56=FNtitle_type(block%,
                 ...variables...)
$(block%+72)=LEFT$(title$,12)
ENDPROC
:
DEFFNflags(...variables...)
LOCAL flag%
flag%=&50
IF titlebar THEN flag%=flag%
                         OR &001
IF movable THEN flag%=flag%
                         OR &002
IF vertical_scroll_bar THEN
             flag%=flag% OR &004
IF horizontal_scroll_bar THEN
             flag%=flag% OR &008
IF NOT back_or_quit_boxes THEN
             flag%=flag% OR &080
=flag%
:
DEFFNtitle_type(...variables...)
LOCAL flag%
flag%=0
IF has_border THEN flag%=flag%
                         OR &004
IF contains_text THEN
             flag%=flag% OR &001
IF text_horizontally_centered
        THEN flag%=flag% OR &008
IF text_vertically_centered
        THEN flag%=flag% OR &010
IF filled_background THEN
             flag%=flag% OR &020
=flag%
```

The program above is an example of how a window might be implemented the procedures/functions with the (...variables...) indicating that any of the variables in that procedure may be passed as parameters, but most will probably be global rather than specific to the individual window. That is some of the variables may be the same for all the windows and so will not need to be passed through the procedure/function, but defined at the beginning of the program, e.g. colours.

'Open_Window' or &400C5

This routine causes a window to appear on the screen at the position (x0,y0,x1,y1) – these are screen coordinates – displaying the contents of the work area extent from (offset_x,offset_y). The scroll bars automatically indicate the position of the visible area relative to the work area extent. The window which appears on the screen is denoted by its handle as previously assigned by the 'Create_ Window' routine. The stack_position is the same as before.

```
DIM open_block%
PROCopen_window(handle%,x0,y0,x1,
y1,offset_x,offset_y,stack_position)
:
END
:
DEFPROCopen_window(handle%,x0,y0,
       x1,y1,offset_x,offset_y,
                stack_position)
LOCAL handle%,x0,y0,x1,y1,
              offset_x,offset_y,
                 stack_position
:
open_block%!0=handle%
open_block%!4=y1
open_block%!8=y0
open_block%!12=x1
open_block%!16=y1
open_block%!20=offset_x
open_block%!24=offset_y
open_block%!28=stack_postion
SYS "Wimp_OpenWindow",
                    open_block%
ENDPROC
```

'Close_Window' or &400C6

This routine causes the specified window to be removed from the screen, NOT from the AWM. If it is required again it may be re-opened. This routine also uses a block to denote the window's handle.

```
DIM handle_block% 1
:
PROCclose_window(handle%)
:
END
:
DEFPROCclose_window(handle%)
LOCAL handle%
?handle_block%=handle%
SYS "Wimp_CloseWindow",
                  handle_block%
ENDPROC
```

'Delete_Window' or &400C3

This routine simply deletes the window from the memory of the AWM. The window is not retrievable once it is deleted! Again it also uses a block to denote the window's handle.

```
DIM handle_block% 1
:
PROCdelete_window(handle%)
:
END
:
DEFPROCdelete_window(handle%)
LOCAL handle%
?handle_block%=handle%
SYS "Wimp_DeleteWindow",
                  handle_block%
ENDPROC
```

In this article we have covered the basic knowledge required to set up a window, display it, remove it from the screen and wipe it from the memory. This in itself is of no use as we cannot yet control the windows in any way! I shall look at the way in which BASIC interacts with the AWM next month, by which time you will, hopefully, be familiar with the techniques of setting up windows!

Using Wordwise Plus

As explained in James Lynn's article last month, Wordwise Plus with its direct screen access on the BBC micro will not work as it stands under the 6502 emulator. However, Computer Concepts have now re-written it so that it will work under the emulator. We look here at its capabilities compared with the BBC version.

I'm sure there are many readers who, like me, have become addicted to Wordwise Plus and have not yet found a substitute for the tremendously powerful text manipulation facilities which its programming language provides. Good news! Computer Concepts have now released a version of Wordwise Plus that will run on the Archimedes, but how does it compare with the BBC version?

Since it is runs under the emulator, it does not make full use of the Archimedes' powerful native mode processing speed. However, since it is effectively a version of Hi-Wordwise Plus, you will find that when you start it up, you will have 43701 bytes free. In terms of the processing speed of the language, sorting 100 addresses into alphabetical order which takes 9.5 minutes on the BBC B compared with 17 minutes on the Archimedes, i.e. only about 55% of the speed of the BBC.

Probably the first thing you will notice when you actually start to use Wordwise Plus on the Archimedes is that the screen display is extremely slow compared with the BBC version. This is partly because it is running under the emulator, but mainly because the mode 7 screen display uses 80k of ram on the Archimedes as against 1k on the BBC micro. Thus up-date the screen requires changes to 80 times as much memory as on the BBC micro. To try to avoid slowing things down too much when you are scrolling through the text, they have used various tricks, the result of which is that the screen display jumps about as you try to move through the document and edit it.

The second thing you will notice is that the shift- and ctrl- cursor movements are the wrong way round! i.e. <ctrl-down> moves you to the end of the text and <shift-right> moves you one word at a time. They have done this deliberately in order to make it the same as View-based products and before you get too hot under the collar, as I did, about pandering to View users, they have provided the facility for changing it back to normal. Key f11 switches between the two conventions!

What of the function keys? The Archimedes, of course, has a different arrangement of function keys to the BBC micro. What they have done is to maintain the different functions according to the fkey numbers, so Command Start is still <f1>, Command End is still <f2> etc. Insert/Overwrite, which was on <f0> has been put onto both f10 and the <print> key (which has taken on the BBC's fkey0 function). This is now not quite so convenient for applications like our own Continuous Processing ROM (for handling large documents) which uses <shift-f0>.

However we are producing CP-ROM on disc (£5 as an up-grade to the ROM version and £20 for a first purchase) and have changed it so that it uses segment 1 instead of segment 0 for the main program. The instinctive <shift-f0> finger reactions can now be used on the <f1> key!

At £10 (+VAT) for the Wordwise Plus upgrade, it is well worthwhile, especially if you have had to sell your BBC to help pay for your Archimedes, but is it worth £29 (+VAT) for the first time purchaser? Well, it depends very much on what the alternatives are. If Acorn's promised free word-processor comes along in December then it might be better to wait, but if you cannot get hold of a copy of View that will work on the Archimedes, Wordwise Plus is a readily available alternative.

One thing worth knowing about Wordwise Plus is that it stores the text as pure ASCII files. This means that you can use it, for example, to prepare programs for Fortram 77, which apparently does not have a built-in editor. You can also use it to prepare and edit boot files and also to edit BASIC programs in a line-number-free environment as explained in "The Complete Wordwise Plus Handbook" (written by Paul Beverley, published by Norwich Computer Services!). One advantage with the Archimedes is that to prepare an existing program for editing without line numbers, all you do is LISTO8 followed by *SPOOL PROGTXT, or whatever, LIST and then *SPOOL at the end of the listing. LISTO8 gives you a listing without line numbers. Then in Wordwise Plus all you do is strip the pad characters off the ends of the lines. (These are linefeed characters which *SPOOL added.) This can be done with a global search and replace changing "|| || ||R" into "||R".

To purchase the disc version of Wordwise Plus, if you have sent in your green registration card to Computer Concepts and can quote your registration number, you can purchase the Archimedes version for just £10 (+VAT). (£10 incVAT through Archive magazine.) For that you get the disc plus a new key-strip. New users will have to pay £29 +VAT but will get a complete set of disc plus keystrip plus manuals. (£30 incVAT through Archive magazine.)

Using BBC ROM images under the emulator

In trying to get BBC ROM software to run under the emulator, there are three distinct problems:

1. How to get a RAM image of a BBC ROM

2. How to get that image to the Archimedes

3. How to get it to run under the emulator

Here, we will only look at the first and the third, since file transfer from 5.25" BBC to 3.5" Archimedes format has been dealt with in various places in Archive magazine.

We will also discuss which ROM's will run under the emulator. We know about a few ROMs that work, but we are hoping that you will supply us with more information.

1. Creating the RAM images

The listing given is a BASIC and machine code program which will copy the contents of the ROMs in the numbered slots in the BBC micro onto the current disc. It asks you which ROM numbers you want to copy - FORM? and TO? – and then proceeds to read them and save them onto disc. Remember that the ROM's are mostly 16k each, so you won't get more than 6 on a 100k disc or 12 on a 200k. The names under which they are saved are the first seven characters of the ROM title. This was done to ensure that the program works on DFS.

We shall assume now that you have somehow transferred the images across onto 3.5" ADFS. (As a last resort, you can get us to do the transfer for you at £5 per disc – see inside back cover.)

2. Getting the emulator working

The best way to do this is to copy the 6502 emulator module across onto the disc that has the images on it, assuming there is enough room. This can be done by typing *COPY :0.$.M*.65* :0.$.65arthur PQ and putting the Welcome disc in when prompted for the source disc and the other disc when asked for the destination disc.

To try out the ROMs, first of all type *INFO $.* to check that the ROM images have load and execution addresses of 008000 (the middle columns where the date stamps of other files appear). If the middle two columns read anything other than 008000 then you should

```
 10 CLS
 20 A%=0
 30 DIM PROG 200
 40 P%=PROG
 50 [OPT 2
 60 STA &FE30
 70 LDA #64
 80 STA &74
 90 LDA #&0
100 STA &70
110 STA &72
120 LDA #&80
130 STA &71
140 LDA #&30
150 STA &73
160 LDY #0
170 .loop
180 LDA (&70),Y
190 STA (&72),Y
200 INY
210 BNE loop
220 INC &71
230 INC &73
240 DEC &74
250 BNE loop
260 LDA &F4
270 STA &FE30
280 RTS
290 ]
300 INPUT"From",S%
310 INPUT"To",E%
320 FOR A%=S% TO E%
330   CALL PROG
340   PRINT"ROM ";A%;"  ";
350   F$=""
360   FOR M%=&3009 TO &300F
370     C%=?M%
380     IFC%>64 F$=F$+CHR$C%
390   NEXT M%
400   PRINT F$
410   OSCLI("SAVE "+F$+
     " 3000 +4000 8000 8000")
420 NEXT A%
```

change it by doing *LOAD IMAGE 10000 and *SAVE IMAGE 10000 +4000 8000 8000.

Now QUIT from BASIC (though you may find that it is not necessary to do so) and type *65ARTHUR or just *6* assuming this is a unique abbreviation. Then type *IMAGE, or whatever names you have used, and this should run the RAM image.

3. Which ROM's work on the emulator?

Test the facilities as much as you can and report to us, if you will, about what sort of success you have had. We can then publish a list and this will save other folk time and effort, working abortively on a ROM that will not work at all under the emulator. Having said that, sometimes one person will be unable to get a particular image working whereas others may be successful. For example, I had problems getting Viewsheet B1.0 working, but I am assured by one reader that the 1.0 version does work – unless perhaps 1.0 is different from B1.0. Any ideas anyone?

The version of View that I used was B3.11 and someone else wrote and said that they had managed to get version 2.1 working. The only other ROMs that I know about so far are LOGOsoft's Master Logo, Viewstore 1.1 and Watford's Modem '84. Viewstore 1.1 apparently does not work as it stands. However, if you *LOAD VIEWSTORE 10000 and poke one byte with ?&1ABE2=&EA and then *SAVE VIEWSTORE 10000 +4000 8000 8000, it will run OK. Not all the features of Modem '84 work under the emulator, but at least it will give you access to Prestel. A

Programmers' Challenge

Here is a challenge for those of you like a good programming problem. Write the shortest possible procedure/function that will return the day of the week ("Mon", "Tue" etc) from a string containing the date and year in the format: "21, Oct 1987". The winner will be the shortest and fastest routine. **Entries submitted on disc, please, by Wednesday 18th November.**

Prize: The winning answer will be published in next month's magazine and the winner will become an Honorary Life Member of the Technical Help Service!!! A

*HELP Information arranged in alphabetical order

***Access** changes the attributes of objects matching the wildcarded specification.
Syntax: *Access <object> [<attributes>]

***Adfs** selects the ADFS as the current filing system.
Syntax: *Adfs

***Append** opens an existing file and subsequent lines of keyboard input are appended to it, input being terminated by ESCAPE.
Syntax: *Append <filename>

***Audio** command controls the sound system
Syntax: *Audio ON | OFF

***Back** swaps current and previous directories.
Syntax: *Back

***Backup** copies one whole floppy disc, (except free space) to another.
Syntax: *Backup <source drive> <dest. drive> [Q]

***BASIC** is the ARM BBC BASIC interpreter.
Syntax: *BASIC [-help|-chain|-load|-quit] <filename>

***Build** opens a new file and subsequent lines of keyboard input are directed to it, input being terminated by ESCAPE.
Syntax: *Build <filename>

***Bye** closes all files, unsets all directories, and parks hard discs.
Syntax: *Bye

***Cat** lists the objects in a directory (default is current).
Syntax: *Cat [<directory>]

***CDir** creates a directory of given name (and size on Net only).
Syntax: *CDir <directory> [<size in entries>]

***ChannelVoice** command attaches a Voice to a Sound Channel
Syntax: *ChannelVoice <channel> <voice index>|<voice name>

***Close** (no parameters) closes all files on the current filing system.

***Compact** tries to collect free spaces together by moving files.
The Q(uick) option will overwrite application workspace.
Syntax: *Compact [<disc spec.>] [Q]

***Con. SoundDefault** sets default sound channel one parameters
SoundDefault <0|1> <0-7> <1-16> (speaker, volume, voice)

***Configure** <item> [<parameter>] sets the CMOS RAM options.

***Copy** copies one or more files that match the given wildcarded specification between directories.

Options:

C(onfirm)	Prompt for confirmation of each copy
D(elete)	Delete the source after copy
F(orce)	Force overwriting of existing objects
P(rompt)	Prompt for the disc to be changed as needed in copy
Q(uick)	Allow use of application workspace to speed file transfer
R(ecurse)	Copy subdirectories and contents
V(erbose)	Print information on each file copied

Syntax: *Copy <source spec> <destination spec> [C][D][F][P][Q][R][V]

***Create** reserves space for the named file, optionally giving it load/exec addresses. No data is transferred to the file.

Synax: *Create <filename> [<size> [<exec addr> [<load addr>]]]

***Delete** tries to delete the named file, giving an error if the file does not exist.

Syntax: *Delete <filename>

***Dir** selects a directory as the current directory (default is &).

Syntax: *Dir [<directory>]

***Dismount** closes files, unsets directories and parks the given disc.

Syntax: *Dismount [<disc spec.>]

***Drive** sets the default drive to use if the directory is unset.

Syntax: *Drive <drive>

***Dump** displays the contents of the file as a hex and ASCII dump.

Syntax: *Dump <filename> [<file offset> [<start address>]]

***Echo** <string> types the string, after transformation.

***EnumDir** creates a file of filenames from a directory that match the supplied wildcarded specification (default is *).

Syntax: *EnumDir <directory> <output ile> [<pattern>]

***Error** generates an error with the given number and text.

Syntax: *Error <number> <text>

***Eval** evaluates an expression.

Syntax: *Eval <expression>

***Ex** lists the objects in a directory together with their file information (default is current).

Syntax: *Ex [<directory>]

***Exec** <filename> directs the operating system to take further input from the given file.

*Exec with no filename causes the exec file to be closed.

Syntax: *Exec [<filename>]

***Format** prepares a floppy disc for use with the DFS.

L is the old 640K ADFS format.

D is a new 800K format.

Syntax: *Format <disc spec.> L/D

***Free** displays the total free space on a disc.

Syntax: *Free [<disc spec.>]

***FX r0 [r1 [r2]]** calls OsByte.

***Go [address] [; environment]** - go to addres; default &8000; text after ';' is environment string.

***Gos** enters the supervisor. Use *Quit to exit.

***HardCopyFX** is used to print a hard copy of the screen.
Syntax: *HardCopyFX [Landscape [<XScale> [<YScale> [<Margin> [<Threshold>]]]]]

***Help** <subjects> atempts to give useful information on the selected topics.
Special keywords include:

Commands	List all the available utility commands
FileCommands	List all the filing system-specific commands
Modules	List the module titles
Syntax	Explain the syntax message format

***Ignore** sets the printer ignore character.
Syntax: *Ignore [<number>]

***Info** lists the file information of an object.
Syntax: *Info <object>

***Key** sets the function keys.
Syntax: *Key <keynumber> [<value>]

***LCat** lists the objects in a subdirectory of the library (default is current library).
Syntax: *LCat [<directory>]

***LEx** lists the objects in a subdirectory of the library together with their file information (default is current library).
Syntax: *LEx [<directory>]

***Lib** selects a directory as the current library.
Syntax: *Lib <directory>

***List** displays the contents of the file in the configured GSRead format. Each line is preceded with a line number.
Syntax: *List <filename>

***Load** with no specified address loads the named file at its own load address. If a Load address is specified, it will be used instead.
Syntax: *Load <filename> [<load addr>]

***Map** displays a disc's free space map.
Syntax: *Map [<disc spec.>]

***Mount** sets the directory to the root directory of the disc, sets the library if unset to $.Library if it exists, and unsets the URD. The default is the default drive.
Syntax: *Mount [<disc spec.>]

***NameDisc** alters a disc's name.
Syntax: *NameDisc <disc spec.> <disc name>

***NoDir** unsets the current directory.
Syntax: *NoDir

***NoLib** unsets the library.
Syntax: *NoLib

***NoUrd** unsets the URD.
Syntax: *NoUrd

***Opt** controls filing system actions.

Opt 1 <n>	Set the filing system message level
Opt 4 <n>	Set the filing system boot option

Syntax: *Opt [<x> [[,] <y>]]

***Pointer** Turns mouse pointer on/off

***Print** displays the contents of a file by sending each byte to the VDU.
Syntax: *Print <filename>

***QSound** queues a sound after the specified numberof tempo ticks
Syntax: *QSound <chan> <amp> <pitch> <duration> <nTicks>

***Remove** tries to delete the named file without causing an error.
Syntax: *Remove <filename>

***Rename** changes the name of an object.
Syntax: *Rename <object> <new name>

***RMClear** deletes all relocatable modules.
Syntax: *RMClear

***RMKill** kills and deletes a relocatable module.
Syntax: *RMKill <ModuleTitle>

***RMLoad** loads and initialises a relocatable module.
Syntax: *RMLoad <FileName>

***RMReInit** reinitialises a relocatable module, reversing the action of *Unplug if appropriate.
Syntax: *RMReInit <ModuleTitle>

***RMRun** runs a relocatable module.
Syntax: *RMRun <FileName>

***RMTidy** compacts and garbage collects the RMA.
Syntax: *RMTidy

***Run** loads and executes the named file, passing optional parameters to it.
Syntax: *Run <filename> [<parameters>]

***Save** copies the given area of memory to the named file.
Syntax: *Save <filename> <start addr> <end addr> [<exec addr> [<load addr>]]
or *Save <filename> <start addr> + <length> [<exec addr> [<load adr>]]

***SChoose** selects a sprite.
Syntax: *SChoose <name>

***SCopy** makes a copy of a sprite.
Syntax: *SCopy <oldname> <newname>

***ScreenLoad** loads into the graphics window.
Syntax: *ScreenLoad <filename>

***ScreenSave** saves the graphics wndow.
Syntax: *ScreenSave <filename>

***SDelete** deletes sprites.
Syntax: *SDelete <name> [<name>]

***Set** assigns a string-type value to a variable.
Syntax: *Set <VarName> <Value>

***SetEval** evaluates an expression and assigns it to a variable.
Syntax: *SetEval <VarName> <Expression>

***SetMacro** assigns a macro-type value to a variable.
Syntax: *SetMacro <VarName> <Value>

***SetType** sets the file type of the named file.
Syntax: *SetType <filename> <filetype>

***SFlipX** reflects the sprite about the X axis.
Syntax: SFlipX <name>

***SFlipY** reflects the sprite about the Y axis.
Syntax: *SFlipY <name>

***SGet** picks up an area of the screen as a sprite.
Syntax: *SGet <name>

***Shadow** makes subsequent mode changes use the other screen bank.
Use *Shadow [0|1]

***Show** lists variabes matching the name given.
Syntax: *Show [<VariableSpec>]

***Shut** closes all open files on all filing systems.
Syntax: *Shut

***ShutDown** closes all open files on all filing systems, logs off file servers and causes hard discs to be parked.
Syntax: *ShutDown

***SInfo** prints size of sprite workspace.
Syntax: *SInfo

***SList** lists all sprites.
Syntax: *SList

***SLoad** loads a sprite file into memory.
Syntax: *SLoad <filename>

***SMerge** appends a sprite file to those in memory.
Syntax: *SMerge <filename>

***SNew** clears all sprite definitions.
Syntax: *SNew

***Sound** command makes a oreground sound
Syntax: *Sound <chan> <amp> <pitch> <duration>

***Speaker** command controls the loudspeaker
Syntax: *Speaker ON | OFF

***Spool** <filename> opens a new file and causes subsequent VDU output to be directed to it, subject to the current *fx 3 status. *Spool with no filename causes the spool file to be closed.
Syntax: *Spool [<filename>]

***SpoolOn** <filename> opens an existing file and causes subsequent VDU output to be appended to it, subject to te current *fx 3 status. *SpoolOn with no filename causes the spool file to be closed.
Syntax: *SpoolOn [<filename>]

***SRename** renames a sprite.
Syntax: *SRename <oldname> <newname>

***SSave** saves the sprite memory.
Syntax: *SSave <filname>

***Stamp** sets the datestamp on a file. If the file was not already datestamped then it is given file type &FFD.
Syntax: *Stamp <filename>

***Status** [<item>] shows the CMOS RAM settings.

***Stereo** sets stereo position of a sound channel
Syntax: *Stereo <chan> <pos>
where <chan> is 1-8, <pos> is -127(L)-127(R) (0 for centre)

***Tempo** command sets the system tempo
Syntax: *Tempo <n> (0 - &FFFF, &1000 Default)

***Time** displays the current real time.
Syntax: *Time

***Title** sets the title of the current directory.
Syntax: *Title [<text]

***Tuning** command sets the system tuning
Syntax: *Tuning n <range (1-32767)>

***TV [<vertical position> [<interlace>]]** sets the position of the display on the screen.

***Type** displays the contents of the file in the configured GSRead format.
Syntax: *Type <filename>

***Unplug** stops ROM modules being initialised.
Syntax: *Unplug [<name>]

***Unset** deletes a variable.
Syntax: *Unset <VarName>

***Up** moves the current directory up the directory structure by the specified number of levels.
Syntax: *Up <levels>]

***Urd** sets the user root directory.
Syntax: *Urd [<directory>]

***Verify** checks the whole disc is readable. The default is the current disc.
Syntax: *Verify [<disc spec.>]

***Voices** lists the installed voices and channel allocation

***Volume** command sets the audio channel loudness
Syntax: *Volume n <range 0-127>

***Wipe** deletes one or more files that match the given wildcard specification.
Options:

C(onfirm)	Prompt for confirmation of each deletion
F(orce)	Force deletion of locked objects
R(ecurse)	Delete subdirectories and contents
V(erbose)	Print information on each file deleted

Syntax: *Wipe <file spec> [C][F][R][V]

*HELP information about the debugger module

***BreakClr** removes the breakpoint at the specified address.
If no address is given then all breakpoints are removed
Syntax: *BreakClr [<address>]

***BreakList** lists all the current set breakpoints
Syntax: *BreakList

***BreakSet** sets a breakpoint at the given address
Syntax: *BreakSet <address>

***Continue** starts execution from the breakpoint saved state
Syntax: *Continue

***Debug** gives access to Debugging facilities
Syntax: *Debug

***InitStore** fills user Memory with the specified data or the value &E1000090 (which is an illegal instruction) if no parameter is given
Syntax: *InitStore [<data/register>]

***Memory** displays the values in the Memory in ARM words
Syntax: *Memory [B] <addr1/reg1> [[+]<addr2/reg2>]

***MemoryA** displays and alters the Memory in bytes or words
Syntax: *MemoryA [B] <addr/reg1> [<data/reg2>]

***MemoryI** disassembles ARM instructions
Syntax: *MemoryI <addr1/reg1> [[+]<addr2/reg2>]

***Quit** returns to the last routine which claimed the EXIT handler
Syntax: *Quit

***ShowRegs** displays the ARM registers
Syntax: *ShowRegs

Symbols used in syntax descriptions:
<> mark sections to be filled in, e.g. <file> indicates that a filename should be supplied here.
[] mark optional sections.

One-Liners

In response to my challenge last month for you to write one-line programs, here are a few of the best ones. Some of them, you will find, are too long to enter as a single line of BASIC because of the limit of the keyboard input buffer. Either enter them as more than one line, or use the ARMBE module on the Welcome disc. Many thanks to W.R.Hindle, Nigel Stuart and Matthew Treagus.

```
  1 REM © W.R.HINDLE 1987
 10 MODE0:
    OFF:
    GCOL3,1:
    VDU24,512;416;768;608;:
    CLG:
    Z%=20:
    FOR X%=0 TO 1279 STEP Z%:
    LINE 1279-X%,0,X%,1023:
    NEXT:
    FOR Y%=0 TO 1023 STEP Z%:
    LINE 0,Y%,1279,1023-Y%:
    NEXT:
    VDU26:
    MOVE 508,412:
    MOVE 772,612:
    OSCLI"SGET FRED":
    OSCLI"SCHOOSE FRED":
    GCOL0,1:
    FOR Z%=1 TO 200:
    PLOT &ED,RND(100)*10,
                   RND(80)*10:
    NEXTZ%:
    ON
```

```
  1 REM>N1
 10 MODE12:
    OFF:
    C=1:
    Z=1:
    FOR K=1 TO 23:
    FOR X=1 TO 15:
    GCOL X:
    Z+=3:
    LINE 1279-Z,1023-Z,Z,Z:
    LINE 1279-Z,Z,Z,1023-Z:
    NEXT X,K:
    REPEAT:
    FOR X=1 TO 15:
    COLOUR X,0,255,0:
    OSCLI"FX19":
    COLOUR X,0,0,0:
    NEXT:
    UNTIL 0
```

```
  1 REM>N2
 10 MODE 12:
    OFF:
    REPEAT:
    FOR X=1 TO 15:
    COLOUR X,0,0,255:
    COLOUR X-1,0,0,0:
    GCOL X:
    MOUSE X%,Y%,Z%:
    LINE 600,0,X%,Y%:
    LINE 600,1023,X%,Y%:
    LINE 0,500,X%,Y%:
    LINE 1279,500,X%,Y%:
    NEXT:
    COLOUR 15,0,0,0:
    UNTIL 0
```

```
  1 REM>N3
 10 MODE15:
    OFF:
    REPEAT:
    S%=RND(200):
    X%=RND(1270):
    Y%=RND(1023):
    GCOL RND(64):
    FOR I=1 TO S% STEP9:
    CIRCLE FILL X%,Y%,I:
```

```
   NEXT:
   GCOL 0:
   FOR I=S% TO 1 STEP-3:
   CIRCLE X%,Y%,I:
   NEXT:
   UNTIL 0

 1 REM>M1
10 MODE1:
   REPEAT:
   FOR C=-15 TO 14:
   RR=ABS(C):
   R=RR << 4:
   COLOUR 1,255-R,0,0:
   GCOL 1:
   CIRCLE FILL 640,512,90:
   COLOUR 2,0,R,0:
   GCOL 2:
   CIRCLE FILL 440,512,100:
   COLOUR 3,0,0,R:
   GCOL 3:
   CIRCLE FILL 840,512,100:
   FORT=1 TO 1000:
   NEXT:
   NEXT:
   UNTIL0

 1 REM>M2
10 MODE 12:
   I=500:
   REPEAT:
   I=I-2:
   C=(C+1)AND7:
   GCOL C:
   ELLIPSE FILL 640,512,I,I*2,RADI:
   UNTIL I<5:
   REPEAT:
   FORK=1 TO 7:
   FORJ=1 TO 7:
   COLOUR J,(J+K) MOD 7:
   NEXT:
   A=INKEY(3):
   NEXT:
   UNTIL0

 1 REM>M3
10 MODE 12:
   R=25:
   C=1:
   N=0:
   ORIGIN640,512:
   REPEAT:
   MOUSEX,Y,B:
   VDUABS(B>0)*12:
   C=((C+1)MOD16)+1:
   GCOL C:
   FOR H=1TO15:
   B=(H+N)MOD16:
   COLOUR H,B*16,B*16,64:
   NEXT:
   N=((N+1)MOD16)+1:
   CIRCLE FILLX,Y,R:
   CIRCLE FILL-X,Y,R:
   CIRCLE FILL-X,-Y,R:
   CIRCLE FILLX,-Y,R:
   UNTIL0
```

Zarch is Here!

D.M.Beazley

Well here it is at last!! The game that everyone has been whispering about all over town is finally out. D.J.Braben, following on from his part in producing "Elite" for the BBC Micro, has yet again produced another masterpiece of software games engineering, utilising the incredible speed and enhanced graphics capabilities of the revolutionary Archimedes.

As with "Elite", the user is prompted to press a key to play or be subjected to the computer showing off and gaining a really high score. A revolving "Lander" craft merely demonstrates the speed of the machine, which must be revolving at least three times as fast "in full colour graphics" as the ship at the start of "Elite".

The landscape is beatifully executed. It is designed as a series of sine curves to give a "gently-rolling hill" effect. The sein wave was clipped at regular intervals so as to produce a set of standard size tiles, just like floor tiles, but put to the shape of the hills. The game is in MODE 13, however it still requires *CONFIGURE SCREENsize 20. This is because he used the *SHADOW command, so needs twice as much memory. This is where the operating system uses a shadow bank of memory for the screen as well as the orginal. This way David Braben is able to update the shadow screen (the one not being viewed), so that when he has finished he just switches memory banks. This makes the graphics very smooth, and you didn't see any drawing occuring, the landscape just appears.

Unfortunately, the depth of view is not very good. This is because, so we are told, there had to be a trade-off between speed and depth of view. I think we can live with it!

Everything in the game has a shadow, to enhance the 'height' effect, and most objects, even the smallest of particles, obeys gravity. To add to the effects the whole scene is 'illuminated' from the top left-handside. That is the objects to the top left are lighter in tint than the ones at bottom right.

Once into the game, the action is fast and furious, the idea being to destroy as many aliens as the computer sends at you on a specific sheet before they completely spread a deadly virus all over the surface of the planet. At the end of the sheet you are given a bonus score depending on the amount of surface area as yet uncontaminated by the red virus.

There are six types of craft that you may encounter:-

1. Seeder
2. Drone
3. Mutant (Mutated Drone)... Sounds rather like Defender doesn't it?!
4. Bomber
5. Pest... Believe me they really are pests
6. Fighter

The best way to deal with a Seeder, which incidently does not shoot back, is to shoot them over water before they can do any damage. What these slow craft try to do is land near, or move towards, one of your own radar installations hoping that you are a poor shot and will accidentally destroy the radar, thus leaving you with a big gap in your radar cover, i.e. a black square on the radar map.

Lots of nasty little ships fly at you from all angles and flying above them is the only way of shooting them since you cannot shoot upwards. Quite often you find yourself in a dogfight with at least three ships. Unless you really are confident and a good shot you have to resort to using a smart bomb or one of your guided missiles.

The one ship that really is a challenge to shoot down is a bomber. These fly fast and low, mostly in straight lines, deploying parachute bombs. So you have to be careful or else you will fly into the bombs, which will detonate on contact and you will be off to meet that great "Zarch Hoverplane" maker in the sky. I found the best way to destroy these is to fly directly behind and a little above and shoot them with your laser cannon but this needs lots of practice. The shear concentration needed to destroy one of these usually results in one running out of fuel without realising it.

Suffice it to say that the graphics are superb, with the planet's landscape of rolling hills, houses, huts and foliage but I was disappointed in that the sound capabilities of this machine have not been exploited to their full potential in this game.

In general, I was very pleased with "Zarch". I feel that, at a cost of £19.95, it is a bit on the expensive side but never-the-less it a must for all Archimedes users if only to show off the computer's amazing processing and display capabilities! A

Reading can be fun!

by Tim Beverley

H S Software produce a Bumper Pack Disc consisting of their Reading Packs 1 to 4 originally written for the BBC micro and Master but now in an enhanced version for the Archimedes. One of the potential users of this software, Tim Beverley (aged 8 - nearly 9!) has assessed this Bumper Pack for Archive readers. He writes…

"This is a word games disc. There are eight games on the disc. The names of the games are Splashdown, Fire Fight, Pyramids, Sploosh, Magic E, Break-In, Sortout and Letterbugs. My favourite game is Fire Fight and my least favourite game is Letterbugs. What you have to do for Fire Fight is you have to use the arrows on the right hand side of the keyboard to move the man. The aim of the game is to get the man to get the words to put into the right boxes because if you do not put them in the right boxes then some more fire goes towards the lady which you are trying to rescue. When you put each bit in, it plays a bit of London's Burning and when you have filled all four boxes it plays all of London's Burning and the man carries the lady down to the ground floor."

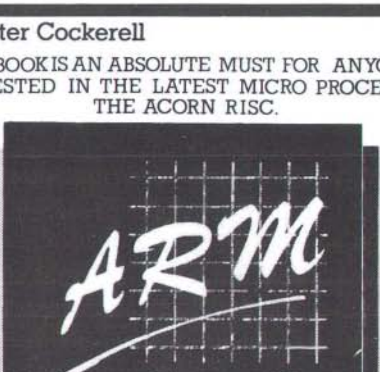
Both Tim and his younger brother, Jonathan (aged 7), have had tremendous fun with these programs. I'm not competent to judge their educational value, but they are great fun to play – even Dad has a go when he gets a chance, usually after the kids are in bed! Their use of graphics is good – basically the same resolution as the BBC version but making use of the greater range of colours available. The sound is good – straightforward four-part harmony on the tunes it plays, but very entertaining. At £19.95, it sounds a bit expensive, but the boys get more play value out of it than Zarch at the same price and it certainly wins over Zarch in terms of educational value!!! HS Software's address is in the FactFile at the back of the magazine.(I'm hoping that HS Software might send us Bumper Pack 2 on review. If not, I might even buy it for Christmas!)

ARM Assembly language programming

By Peter Cockerell

THIS BOOK IS AN ABSOLUTE MUST FOR ANYONE INTERESTED IN THE LATEST MICRO PROCESSOR, THE ACORN RISC.

- **Introduction: A simple introduction to assembly language for beginners**
- **Architecture: Details of the registers, flags, etc.**
- **Instruction set: A complete run-down of the various instructions, conditional execution, stack handling, barrel shifter, load and store, add and subtract . . .**
- **The BBC BASIC Assembler: Particularly useful to Archimedes users, this details the use of the assembler provided within BASIC-V. Covers syntax, expressions, assembly options, multi-pass assembly . . .**



- **Programming principles: General theory and working examples of conditional execution, loops, branches and jump tables, subroutines and parameter passing**
- **Data structures: String handling, searches, comparisons, multiplication and division, ASCII to binary conversion, floating point, arrays, etc.**
- **Hardware: Provides an understanding of each part of the Acorn chip set (MEMC, VIDC, IOC and the processor) as used in the Archimedes**

The only book which covers Acorn RISC Assembly Language

£12.95 incl.

AVAILABLE NOW

Computer Concepts Ltd

Gaddesden Place Hemel Hempstead Herts HP2 6EX Tel 0442 63937

ACCESS & VISA WELCOME

— All orders despatched in 24 hours

Library, Machine Code or Module?

A 4.5" by 3.5" Archimedes screen dump by Gerrald Fitton

(The programs and text of the three articles are ©1987 ABACUS TRAINING, 29 Okus Grove, Upper Stratton, Swindon, SN2 6QA.)

Well, why not take your pick?! This is a three part article which is meant to be both useful and informative. In the article a mini screen dump is used as an example to highlight features of the Archimedes software in a way that may be of use to other programmers. My motivation for developing this mini dump is that one of the rules I teach for illustrative charts (in Applied Statistics) is:- "Keep the chart as small as possible and always put it on the same page as the explanatory text". After patching up the "bug"(see Hints & Tips), the screen dump modules supplied by Acorn (eg HardCopyFX) will output a range of sizes but the smallest is about 9" by 7". This is only just smaller than an A4 page and left me insufficient space for text.

You may want to use the printer dump in your own applications as it stands, you may want to modify it, or you may find more of interest in the discussion than in the program itself.

In the first part of the article, the mini dump is written as a self contained BASIC function. This function can be included either in the user's program by making appropriate adjustments to the line numbers or, better still, it can be loaded into memory using the excellent new LIBRARY or INSTALL commands which are available in BASIC V. In the second part, a program is written in ARM assembler which mimics the BASIC version. This version assembles to fully relocatable ARM machine code, which can be saved to disc and *LOADed from within a BASIC program to any (unknown) location in memory, preferably a local byte array, and then activated using the BASIC CALL command with two parameters. The third part of the article develops the source code further so that when it has been assembled, the code becomes a fully relocatable module. Such a module, once it has been loaded, appears to the user as an extension of the operating system being called with a * command in the usual way.

Specification

• The dump will be about 4.5" by 3.5", actually 40/9" by 32/9".

• It will use the Epson graphics mode 7 which gives 144 dots per inch horizontally and 72 dots per inch vertically. This printer mode maps circles to true circles without elliptical distortion and matches the highest screen resolution available without a multi-synch monitor. In these screen modes (eg mode 0) the screen consists of 640 by 256 pixels and each pixel maps to one printed dot. Epson graphics mode 7 is available on the FX80+ and later machines but not on the FX80 or RX80. Because of this an alternative program for graphics mode 5 is given. In printer mode 5 there is still no elliptical distortion but the horizontal resolution is 72 dots per inch. This reduces the intensity of the printout and faithfully maps only those screen modes with resolution 320 by 256 or coarser.

• It will faithfully dump screens in which the graphics origin has been moved away from the bottom left corner with either the ORIGIN or VDU 29 commands.

• The use of VDU 24 to define a graphics window will cause the screen dump to ignore, and not print, the screen area outside the defined window.

• The printer will leave a left margin which is passed to the function as a parameter and defined in tenths of an inch matching pica style characters.

• The program includes a means of selecting which logical colours map to printed dots. This is to define a threshold logical colour number which is passed as a parameter to the function. If the logical colour of the screen pixel is above this threshold value then a dot is printed.

```
  100 REM > MainProg : An example.
  110 :
  130 REM Author G.L.Fitton.
  140 REM 17th October 1987.
  150 :
  170 ON ERROR:CLOSE#0:VDU4,26,12:OSCLI("FX3"):REPORT:
                                        PRINT " at line ";ERL:STOP
  180 :
  190 MODE 1
  200 :
 1000 LIBRARY "MiniDump7"
 1010 margin%=19
 1020 threshold%=0
 1030 *PRINT picture
 1040 dump$="FNminidump("+STR$(margin%)+","+STR$(threshold%)+")"
 1050 dump%=EVAL(dump$)
 1060 IF dump%=FALSE THEN PRINT "Illegal value of a parameter"
 1080 END
 1090 :

49999 REM > MiniDump5 : A self-contained function.
50000 DEF FNminidump(margin%,threshold%)
50010 REM The BASIC version.
50020 :
50040 IF margin%>79 OR threshold%>255 THEN =FALSE
                                              :REM Bad parameters.
50050 :
50070 LOCAL left%,bottom%,right%,top%,xorigin%,yorigin%,
                                               xconvert%,yconvert%
50080 LOCAL line%,x%,points%,pixel%,printcode%,strike%,
                                                      colour%,vdu%
50090 DIM vdu% 39
50110 :
50120 !(vdu%+ 0)=  4
50130 !(vdu%+ 4)=  5
50140 !(vdu%+ 8)=128
50150 !(vdu%+12)=129
50160 !(vdu%+16)=130
50170 !(vdu%+20)=131
```

```
50180 !(vdu%+24)=136
50190 !(vdu%+28)=137
50200 !(vdu%+32)=-1
50210 SYS "OS_ReadVduVariables",vdu%,vdu%
50220 xconvert%=!(vdu%+ 0)
50230 yconvert%=!(vdu%+ 4)
50240 left%    =!(vdu%+ 8)
50250 bottom%  =!(vdu%+12)
50260 right%   =!(vdu%+16)
50270 top%     =!(vdu%+20)
50280 xorigin% =!(vdu%+24)
50290 yorigin% =!(vdu%+28)
50300 :
50310 :
50320 left% = (left%<<xconvert%)-xorigin%
                                     :REM These are the screen
50330 right% = ((right%+1)<<xconvert%)-xorigin%-1
                                      :REM co-ordinates of the
50340 bottom% = (bottom%<<yconvert%)-yorigin%
                                     :REM the graphics window.
50350 top% = ((top%+1)<<yconvert%)-yorigin%-1
50360 :
50380 points% = (right%-left%+1)DIV4
                          :REM The number of printed points per line.
50390 :
50410 REM *FX 3,10 sends to printer only.
50420 *FX 3,10
50430 VDU 27,64        :REM Initialise the printer.
50440 VDU 27,65,8      :REM 8/72" line spacing = 8 dots per line.
50450 VDU 27,108,margin% :REM Left margin is margin%.
50460 :
50480 FOR line%=top% TO bottom% STEP -32
50490   VDU 27,42,5,points%MOD&100,points%DIV&100:REM Printer
       mode 5.
50500   FOR x%=left% TO right% STEP 4
50510     printcode%=0 :REM Initialise the value sent
                                              to the printer.
50520     FOR pixel%=7 TO 0 STEP -1 :REM Assemble 'colour'
                                                 of 8 pixels.
50530       colour%=POINT(x%,line%-28+(pixel%<<2))
                                    :REM If colour% is greater
50540       printcode%+=(-(colour%>threshold%))<<pixel%
                                     :REM than threshold% then
```

```
50550      NEXT pixel%      :REM print a dot.
50560      VDU printcode%   :REM One printhead strike prints
                                                 a column of 8 dots.
50570   NEXT x%
50580   VDU 10,13           :REM Line feed and carriage return.
50590 NEXT line%
50610 :
50620 VDU 12                :REM Form feed.
50630 VDU 27,64             :REM Reset the printer.
50640 :          REM *FX 3,0 restores output to the screen only.
50660 *FX 3,0
50670 :
50680 =TRUE

49999 REM > MiniDump7 : A self-contained function.
50000 DEF FNminidump(margin%,threshold%)
50010 REM The BASIC version.
50020 :
50030 :
50040 IF margin%>79 OR threshold%>255 THEN =FALSE
                                              :REM Bad parameters.
50050 :
50060 :
50070 LOCAL left%,bottom%,right%,top%,xorigin%,yorigin%,
                                                 xconvert%,yconvert%
50080 LOCAL line%,x%,points%,pixel%,printcode%,strike%,
                                                        colour%,vdu%
50090 DIM vdu% 39
50100 :
50110 :
50120 !(vdu%+ 0)=  4
50130 !(vdu%+ 4)=  5
50140 !(vdu%+ 8)=128
50150 !(vdu%+12)=129
50160 !(vdu%+16)=130
50170 !(vdu%+20)=131
50180 !(vdu%+24)=136
50190 !(vdu%+28)=137
50200 !(vdu%+32)=-1
50210 SYS "OS_ReadVduVariables",vdu%,vdu%
50220 xconvert%=!(vdu%+ 0)
50230 yconvert%=!(vdu%+ 4)
50240 left%    =!(vdu%+ 8)
```

```
50250 bottom%  =!(vdu%+12)
50260 right%   =!(vdu%+16)
50270 top%     =!(vdu%+20)
50280 xorigin% =!(vdu%+24)
50290 yorigin% =!(vdu%+28)
50300 :
50320 left%=(left%<<xconvert%)-xorigin%:REM These are the screen
50330 right%=((right%+1)<<xconvert%)-xorigin%-1
                                         :REM co-ordinates of the
50340 bottom%=(bottom%<<yconvert%)-yorigin%
                                        :REM the graphics window.
50350 top%=((top%+1)<<yconvert%)-yorigin%-1
50360 :
50380 points%=(right%-left%+1)DIV2
                      :REM The number of printed points per line.
50390 :REM *FX 3,10 sends to printer only.
50420 *FX 3,10
50430 VDU 27,64           :REM Initialise the printer.
50440 VDU 27,65,8         :REM 8/72" line spacing = 8 dots per
       line.
50450 VDU 27,108,margin% :REM Left margin is margin%.
50460 :
50470 :
50480 FOR line%=top% TO bottom% STEP -32
50490   VDU 27,42,7,points%MOD&100,points%DIV&100
                                            :REM Printer mode 7.
50500   FOR x%=left% TO right% STEP 2
50510     printcode%=0
                  :REM Initialise the value sent to the printer.
50520     FOR pixel%=7 TO 0 STEP -1
                             :REM Assemble 'colour' of 8 pixels.
50530       colour%=POINT(x%,line%-28+(pixel%<<2))
                                        :REM If colour% is greater
50540       printcode%+=(-(colour%>threshold%))<<pixel%
                                        :REM than threshold% then
50550     NEXT pixel%                   :REM print a dot.
50560     VDU printcode%
            :REM One printhead strike prints a column of 8 dots.
50570   NEXT x%
50580   VDU 10,13          :REM Line feed and carriage return.
50590 NEXT line%
50610 :
```

```
50620 VDU 12              :REM Form feed.
50630 VDU 27,64           :REM Reset the printer.
50640 :
50650 REM *FX 3,0 restores output to the screen only.
50660 *FX 3,0
50670 :
50680 =TRUE
50690 :
```

ARM assembly program for a 'hand' pointer!

As mentioned in the Question and Answer section, Jeff Shipp has given us a hand (!) so that we can point with it to our Archimedes screen – it's quite a realistic hand, especially in mode 9 which the program itself selects.

There is no explanation given as to how this all works, but if you cannot decipher it from the program, Adrian Look is going to be covering it later in his series on the WIMP environment.

```
   10 REM > Hand
   20 REM Hand Assembler program
   30 REM by J Shipp, 1987
   40
   50 REM Declare register numbers
   60 defn   = 0
   70 blkptr = 1
   80 tblptr = 2
   90 index  = 3
  100 sp     = 13
  110 link   = 14
  120 Defpointer = &15 : REM OSWORD to define mouse pointer
                                                 size, shape etc.
  130 DIM code 400      : REM Space for code
  140 FOR pass=0 TO 2 STEP 2
  150   P%=code
  160   [ opt pass
  170   ;
  180   stmfd (sp)!,{link}        ; Save the return address
  190   \-
  200   mov  defn,#Defpointer     ; r0 contains reason code
  210   adr  blkptr,parmBlock     ; r1 points at parm block
  220   adr  tblptr,datatable     ; r2 points at datatable
  230   add  index,blkptr,#5      ; Get index of datatable
                                            pointer in parm block
```

```
240   .loop
250   strb tblptr,[index,#1]!   ; Save each byte of the
                                          datatable pointer
260   movs tblptr,tblptr,lsr #8 ; into parm block
270   bne loop
280   swi "OS_Word"             ; Write new definition.
290   \-
300   ldmfd (sp)!,{pc}          ; Return
310   \-
320   .parmBlock
330   EQUB &00        ;
340   EQUB &02        ; Shape number (1 - 4)
350   EQUB &08        ; Width (w) in bytes (0 - 8)
360   EQUB &20        ; Height (h) in pixels (0 - 32)
370   EQUB &00        ; ActiveX in pixels from left (0 - w*4-1)
380   EQUB &0D        ; ActiveY in pixels from top (0 - h-1)
                                             i.e. fingertip
390   EQUB &00        ; Least significant byte of pointer to
                                                   datatable
400   EQUB &00        ;
410   EQUB &00        ;
420   EQUB &00        ; Most significant byte of pointer to
                                                   datatable
430   \-
440   .datatable
450   ;                                   Row
460   ;
470   EQUD &00000000:EQUD &00000000 ;  1
480   EQUD &00000000:EQUD &00000000 ;  2
490   EQUD &00000000:EQUD &00000000 ;  3
500   EQUD &00000000:EQUD &00000C00 ;  4
510   EQUD &00000000:EQUD &00003D00 ;  5
520   EQUD &00000000:EQUD &0000F500 ;  6
530   EQUD &00000000:EQUD &00015500 ;  7
540   EQUD &00000000:EQUD &00055500 ;  8
550   EQUD &00000000:EQUD &00155400 ;  9
560   EQUD &00000000:EQUD &00555000 ; 10
570   EQUD &00000000:EQUD &01554000 ; 11
580   EQUD &00000000:EQUD &05550000 ; 12
590   EQUD &55555554:EQUD &15555555 ; 13
600   EQUD &55555555:EQUD &55555555 ; 14
610   EQUD &55555555:EQUD &55555555 ; 15
620   EQUD &55555554:EQUD &55555555 ; 16
```

```
630    EQUD &90000000:EQUD &555FEAAA ; 17
640    EQUD &54000000:EQUD &557FF555 ; 18
650    EQUD &54000000:EQUD &555FD555 ; 19
660    EQUD &50000000:EQUD &55559555 ; 20
670    EQUD &00000000:EQUD &556AAAAA ; 21
680    EQUD &40000000:EQUD &57FD5555 ; 22
690    EQUD &40000000:EQUD &57FD5555 ; 23
700    EQUD &40000000:EQUD &55655555 ; 24
710    EQUD &00000000:EQUD &555AAAA5 ; 25
720    EQUD &00000000:EQUD &557D5550 ; 26
730    EQUD &00000000:EQUD &55FF5550 ; 27
740    EQUD &00000000:EQUD &057D5550 ; 28
750    EQUD &00000000:EQUD &015A5540 ; 29
760    EQUD &00000000:EQUD &00000000 ; 30
770    EQUD &00000000:EQUD &00000000 ; 31
780    EQUD &00000000:EQUD &00000000 ; 32
790 ]:NEXT
800 CALL code
810 MODE9
820 *POINTER
830 MOUSE COLOUR 1,225,175,162
840 MOUSE COLOUR 2,209,139,140
850 MOUSE COLOUR 3,215,0,0
860 MOUSE ON 2
```

Mode 7 Screen-save Program

Having read that mode 7 requires 80K of RAM and that Arthur decides where screen memory should reside depending on the circumstances, and having discovered that *SCREENSAVE does not work in mode 7 anyway, I decided that I needed a couple of procedures to save the mode 7 screen as a file of characters. This means that 80K of screen RAM can be saved as a file of 1K. The procedures are called load_screen and save_screen and I have built them into an example program:

```
 10 REM Procedure to save MODE 7 screen
 20 REM Clive Williams 15.8.1987
 30
 40 OS_Byte=6
 50 Read_char=135
 60
 70 MODE 7:OFF
 80 loadfile$="DEMO":savefile$="DEMO"
 90 REPEAT
100   CLS
110   FOR y%=1 TO 2
120     PRINTTAB(0,y%);CHR$157;CHR$129;CHR$141;"Using";
130     PRINTCHR$132;"BPUT";CHR$129;"and";CHR$132;"BGET";
140     PRINTCHR$129;"for";CHR$130;"MODE 7";CHR$129;
                                          "files";CHR$156
150   NEXT y%
160   PRINTTAB(10,4);CHR$131;"Which option?"
170   PRINTTAB(10,6);CHR$134;"1.Save screen"
180   PRINTTAB(10,7);CHR$134;"2.Load screen"
190   PRINTTAB(10,8);CHR$134;"3.Finish"
200   PRINTTAB(9,10);CHR$131;"Select 1,2 or 3"
210   REPEAT
220     choice=GET-48
230   UNTIL choice>0 AND choice<4
240   CASE choice OF
250     WHEN 1 : PROCcreate_screen
260     PROCsave_screen(savefile$)
270     WHEN 2 : PROCload_screen(loadfile$)
280     G=GET
290   ENDCASE
300 UNTIL choice=3
310 ON:CLS
320 END
330
340 DEFPROCcreate_screen
```

```
350 LOCAL x%,y%
360 CLS
370 FOR y%=0 TO 23:PRINTTAB(0,y%);CHR$132;CHR$157;:NEXT
380 FOR y%=0 TO 1:P.TAB(10,y%);CHR$135;CHR$141;"Example
     Screen";:NEXT
390 PRINTTAB(3,3);CHR$131;"This is a sample text screen to"
400 PRINTTAB(3,4);CHR$131;"show how BPUT and BGET can be"
410 PRINTTAB(3,5);CHR$131;"used to save MODE 7 screens as"
420 PRINTTAB(3,6);CHR$131;"files of 1K."
430 P.TAB(3,8);CHR$134;"The command";CHR$129;"*SCREENSAVE";
     CHR$134; "can't"
440 PRINTTAB(3,9);CHR$134;"be used as MODE 7 is not classed"
450 PRINTTAB(3,10);CHR$134;"as a graphics mode."
460 PRINTTAB(3,12);CHR$131;"Although Archimedes stores MODE 7"
470 PRINTTAB(3,13);CHR$131;"screens using 80K of RAM, this"
480 PRINTTAB(3,14);CHR$131;"program shows how to reduce this"
490 PRINTTAB(3,15);CHR$131;"to 1K for file storage."
500 FOR y%=17 TO 19:PRINTTAB(3,y%);CHR$(145+(y%-17));
510   FOR x%=160 TO 191:VDU x%:NEXT x%
520 NEXT y%
530 FOR y%=20 TO 22:PRINTTAB(3,y%);CHR$(145+(y%-16));
540   FOR x%=160 TO 191:VDU x%:NEXT x%
550 NEXT y%
560 PRINTTAB(8,23);CHR$133;"Press a key to continue";
570 ENDPROC
580
590 DEFPROCload_screen(file$)
600 LOCAL char,channel,x%,y%
610 CLS
620 channel=OPENIN file$
630 FOR y%=0 TO 23
640   FOR x%=0 TO 39
650     VDU BGET#channel
660   NEXT x%
670 NEXT y%
680 CLOSE#channel
690 ENDPROC
700
710 DEFPROCsave_screen(file$)
720 LOCAL char,channel,x%,y%
730 channel=OPENOUT file$
740 FOR y%=0 TO 23
```

```
750     FOR x%=0 TO 39
760       PRINTTAB(x%,y%);
770       SYS OS_Byte,Read_char
      TO ,char
780       BPUT#channel,char
790     NEXT x%
800 NEXT y%
810 CLOSE#channel
820 ENDPROC
```

Most of this program is self-evident, however I offer a few notes for the benefit of those new to the Archimedes.

The keyword OFF, on line 70, turns the cursor off. It is turned on again, at the end of the program, by the ON on line 310. The CASE statement between lines 240 and 290 removes the need for IF...THEN or an ON...PROC structure such that when option 1 is selected, the program creates a sample screen and then saves it, or if option 2 is selected the sample screen is loaded and displayed and the program waits for a key to be pressed. No action is taken for option 3 as it is caught by the UNTIL on line 300 and the program ends.

PROCcreate_screen simply creates the sample screen ready for saving.

PROCsave_screen legally peeks each screen position using *FX 135 which returns the character in R1 (and the screen mode, if required, in R2). I have encoded this call using the keyword SYS on line 770. The variables OS_Byte and Read_char are set to 6 and 135 respectively, on lines 40 and 50. OS_Byte is set to 6 because osbyte is SWI code 6 and *FX commands are calls to osbyte. Thus SYS OS_Byte, Read_char is the same as *FX 135.

The keyword TO tells the operating system what to do with the returned values : the comma means "ignore the contents of R0" (ie. register 0) and char means "put the value in R1 into the variable char". The other registers are ignored. To ignore everything except the current mode, the line would be:

SYS OS_Byte,Read_char TO ,,mode

That is, ignore R0 and R1 and store R2 in the variable called mode.

Once the character at the current cursor position has been read, it is saved to the file with BPUT. Even using this slow method of file saving, Archimedes works fast enough to allow it to be acceptable : the file is saved in 1.7 seconds!

PROCload_file legally displays the file, regardless of the actual screen memory address, by reading each byte from the file and displaying it with VDU. Again, at 1.3 seconds, this operation is acceptably fast.

Order Form

		Quantity	Total
File Transfer Service:	Send your 5.25" discs, DFS or ADFS and £5 per disc, inclusive, and we will transfer the files onto an ADFS 3.5" disc. (One 3.5" for each 5.25".)		
Program Disc	Disc of the programs in this issue 1 – £3		
	Disc of the programs in this issue 2 – £3		
Blank Keystrips	A4 cards with 5 blank keystrips on each – £1 per card		
Custom Printed Keystrips	for your own application. Layout similar to the BASIC and Sprite Editor keystrips – £7 each (£5 if text supplied on disc).		
Unformatted 3.5" Discs	Bulk pack, unbranded – 10 for £16		
	Wabash in library cases – 10 for £24		
Wordwise Plus – Full Package £30			
Wordwise Plus – Upgrade £10 (Please quote registration number.)			
"ARM Assembly Language Programming" £12			
Clares' Archimedes Toolkit Module – £37			
Clares' Artisan Art Package – £37			
Programmers Reference Manual (Ring for price and availability.)			

I enclose a cheque payable to "Norwich Computer Services" for: ______

All prices are inclusive of VAT and UK carriage.

Name ______________________________

Address ______________________________

____

____

*HELP Archive

What are its aims? – Archive is a subscription magazine, for users of the Acorn Archimedes range of computers, which aims (1) to provide information to the user (2) to provide a forum in which we can all share ideas (3) to give the benefit of bulk buying of software (4) to allow software and hardware vendors to advertise their wares.

Is it a User Group? – No, but I would like it to have a "User Group feel" – like BEEBUG when it first started – except that Norwich Computer Services has to earn a living from the magazine – hence the order form overleaf.

However, Sue and I are not in this business to make lots of money; we enjoy the work we do and it's very satisfying to be able to provide a useful service. We only ask to make enough money to live on plus a bit to give away and then we'll be happy. (I hope it doesn't sound too trite, because it's true.)

HELP! – Could you help us, please, to make Archive a success? We don't have the muscle or the financial budget of the big magazines and cannot afford to do a vast amount of advertising, so, if you think Archive is good, please take out a full subscription (if you have not already done so) and recommend the magazine to a friend or two. If you want any more information sheets and subscription forms, please let us know.

Can we answer technical enquiries by phone? As much as we would like to continue the policy we have always had of being available to answer all your technical enquiries by phone, we felt that if we were not careful, we would be so inundated with calls that we would not have time to get the information out to you through the magazine. For that reason, we have introduced the Technical Help Service so that those who really need the instant access to help can purchase it. (£8 per year.) I hope you will bear with us and, if you do not feel it is worth the extra £8, please send your enquiries by post.

Do we take Access or Visa? No, I'm afraid not – mainly because of the high percentage that we would have to pay for the privilege (I think it's 6% when you first start). The other reason is that we have always had a policy of sending goods out by return of post, whenever possible, regardless of whether the cheques have cleared through the bank. The way we see it is that if you trust us by sending a cheque without the Access guarantee, it is reasonable for us to trust you by sending out the goods without waiting for the cheque to clear. Perhaps we are foolish to take the risk, but we have only had two dud cheques in three years of full time trading.

Although there's only two of us here, we are surrounded by a lot of people who help us in various ways, some voluntarily and some professionally, without whom we would not be able to stay in business. I don't think it is appropriate to mention them all by name, but I would just like to assure them that we are really grateful to them all. However, as I have said in most of my previous publications, as a committed Christian, there is One Person who never fails us and without whom we would achieve nothing worthwhile. God supplies all our needs, and we thank and praise Him!

I hope you find Archive interesting and informative and that you will feel able to contribute your own ideas, hints and tips or even full articles. I'm afraid that we cannot afford to pay the vast sums for articles, but at least you would automatically become an "HLMTHS". – Honorary Life Member of the Technical Help Service! A

Fact-File

Brainsoft	22 Baker Street, London, W1M 1DF.
CJE Micros	Dept AM, 78 Brighton Road, Worthing, W Sussex, BN11 2EN. (0903-213361)
Clares Micro Supplies	98 Middlewich Road, Rudheath, Northwich, Cheshire, CW9 7DA. (0606-48511)
Computer Concepts	Gaddesdon Place, Hemel Hempstead, Herts, HP2 6EX. (0442-63933)
HS Software	56, Hendrefolian Avenue, Sketty, Swansea, SA2 7NB. (0792-204519)
Intelligent Interfaces	14 Julius Close, Chandlers Ford, Eastleigh, Hants, SO5 2AB. (04125 61514)
LOGOsoft	316a Richmond Road, Twickenham, Middlesex, TW1 2PD. (01-891-0989)
Logotron	(0223-323656)
Meadow Computers	11, London Street, Whitchurch, Hants, RG28 7LH. (025689-2008)
Minerva Systems	69 Sidwell Street, Exeter, EX4 6PH. (0392 - 37756)
Superior Software	Regent House, Skinner Lane, Leeds, LS7 1AX. (0532-459453)

User Guide Errata

Let us know if you find any mistakes in the User Guide. We can then keep everybody informed.

• p33	It looks as if a whole section is missing as the program example stops in the middle.
• p87 line1	"It does not give no information" – oops!
• *p90*	*Indented para starting "NOTE:" for "*CONFIGURE 5" read "*CONFIGURE FILE 5"*
• *p135*	**SSPACE 20 – no reference or explanation. Anyone got any ideas?*
• *p143*	*The code for red in the table is 129, not 29*
• p165 middle	All the ">>2" and ">>>2" should be ">>1" and ">>>1"
• p405last line	"shift-f2 (UNDO)" should be "shift-f10 (UNDO)"
• p408	The final statement is incorrect: <ctrl-R> **cannot** be used to remove line markers (well not on my version of ARMBE, anyway.)

(Errata in italics were added to the list since the last issue of Archive.)

Archive

The Subscription Magazine for *Archimedes* users

Articles in previous issues:

- File transfer on RS423
- Attaching a 5.25" drive
- C.C.'s ROM-Link packages
- Clares' Toolkit module
- Graphics demo programs
- Archie the Music Computer
- Structuring with BASIC V
- Technical notes on BASIC V
- Epson Screen Dump
- Using BBC ROM images

Future issues will include:
BBC micro as an I/O podule, using windows, using the operating system, ARM assembly language programming, writing relocatable modules, etc etc.

Technical Help Service (£8 / year)
A telephone hot-line service for immediate help with your technical problems. Any member can send written enquiries, but for a fast response use the THS!

Members discount: 7.5% off software from Computer Concepts, Minerva Systems and Clares Micros Supplies purchased through Norwich Computer Services.

Subscription: 12 issues £10 (UK/Eire) Europe £16, Middle East £20, America / Africa £23, Elsewhere £25. Technical Help Service £8

* Please send copies of *Archive* magazine for one year starting from
Vol. 1, Nº 1, Oct '87 / Vol. 1, Nº 2, Nov '87 / Vol. 1, Nº 3, Dec '87.

* Please enrol me on the Technical Help Service for one year. (£8)

I enclose a cheque for £ ________

Name: ____________________

Address: ____________________

____

____________________ Postcode: __________

Norwich Computer Services, 18 Mile End Road, Norwich, NR4 7QY.
Subscription – £10 per year, Technical Help Service – £8 per year